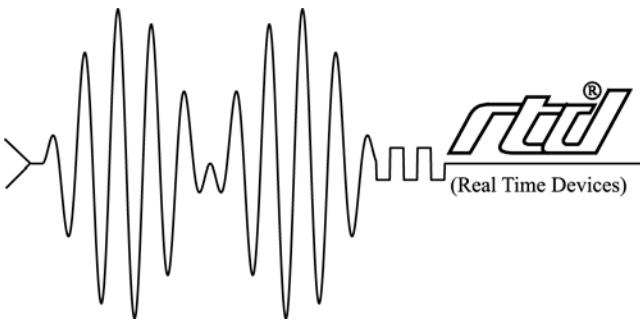
CMX47786HX
cpuModule™ User's Manual

RTD Enhanced Award BIOS Versions 6.00.xx



RTD Embedded Technologies, Inc.
"Accessing the Analog World"®

BDM-610000038
Rev. A

CMX47786HX
cpuModule™
User's Manual

RTD Embedded Technologies, Inc.
103 Innovation Blvd.
State College, PA 16803-0906

Phone: +1-814-234-8087
FAX: +1-814-234-5218

E-mail
sales@rtd.com
techsupport@rtd.com

web site
http://www.rtd.com

Revision History

Rev A Initial Release

Published by:

RTD Embedded Technologies, Inc.
103 Innovation Blvd.
State College, PA 16803-0906

Copyright 1999, 2002-2005 by RTD Embedded Technologies, Inc.
All rights reserved
Printed in U.S.A.

The RTD Logo is a registered trademark of RTD Embedded Technologies. cpuModule and utilityModule are trademarks of RTD Embedded Technologies. PS/2, PC/XT, PC/AT and IBM are trademarks of International Business Machines Inc. MS-DOS, Windows, Windows 95, Windows 98 and Windows NT are trademarks of Microsoft Corp. PC/104 is a registered trademark of PC/104 Consortium. All other trademarks appearing in this document are the property of their respective owners.

Chapter 1: Introduction 3

The CMX47786HX cpuModule 4
Specifications 6

Chapter 2: Getting Started 9

Basic Connector Locations 10
Fail safe boot ROM 11
Cable Kits 12
Connecting Power 13
Connecting the utility cable 14
Connecting a Keyboard 15
Connecting to the PC/104 Bus 16
Connecting to the PC/104-Plus PCI Bus 17
Booting the cpuModule for the First Time 19
Booting to Boot Block Flash with Fail Safe Boot ROM 20
If You Misconfigure the cpuModule 21
For More Information 22

Chapter 3: Connecting the cpuModule 23

Connector Locations 24
Auxiliary Power CN3 26
Serial Ports, CN7 and CN8 28
multiPort, CN6 - Parallel Port Mode 31
multiPort, CN6 - aDIO Mode 32
multiPort, CN6 - Floppy Controller Mode 33
Multifunction Connector, CN5 35
VGA Video Connector, CN18 37
LVDS Flat Panel Video Connector, CN19 38
EIDE Connector, CN9 39
Bus Mouse Connector, CN4 41
USB Connector,CN17 42
10/100 Base T and TX Connector, CN20 43
Audio, CN11 44
PC/104 Bus, CN1 and CN2 45
PC/104-Plus PCI Bus, CN16 49

Chapter 4: Configuring the cpuModule (BIOS Setup) 53

Disabling Fail Safe Boot ROM 54
Installing SSD Memory 55
Configuring with the RTD Enhanced Award BIOS 56

Chapter 5: Using the cpuModule 57

Memory Map 58
Input/Output Address Map 59
Hardware Interrupts 61
The RTD Enhanced Award BIOS 63
Thermal Throttling 64

Direct Hardware Control 65
Advanced Digital I/O Ports (aDIO) 66
Basic Interrupt Information for Programmers: 72
Watchdog Timer Control 75
Multi-Color LED 78
Real Time Clock Control 79
Parallel Port Control 80
Storing Applications On-board 81
Ways to Use the Solid State Disk Sockets 82
Using BIOS Extension Devices 83
Using Conventional Solid State Devices 84

Chapter 6: Hardware Reference 87

Jumpers and Solder Jumper Settings 88
Mechanical Dimensions 91
Processor Thermal Management 92

Chapter 7: Troubleshooting 93

Common Problems and Solutions 94
Troubleshooting a PC/104 System 97
How to Obtain Technical Support 98

Chapter 8: Limited Warranty 99

Table Of Figures

CMX47786HX Block Diagram 5
CMX47786HX Basic Connector Locations 10
CMX47786HX Connector Locations 24
CMX47786HX Jumper Locations - Top 89
CMX47786HX Jumper Locations - Bottom 89

Table Of Tables

SSD Conventional Devices 7
SSD BIOS Extension Devices 7
Power Consumption 8
CMX47786HX Connectors 25
Auxiliary Power Connector CN3 26
Serial Port Base Address and IRQ Options 28
Serial Port in RS-232 Mode 28
RS-422 Serial Port Connections 29
RS-485 Serial Port Connections 29
Serial Port in RS-422/485 Mode 30
multiPort Parallel Port Connector Pinout CN6 31
multiPort Advanced Digital I/O Connector Pinout CN6 32
multiPort Connector Floppy Pinout CN6 33
Multifunction Connector CN5 35
Keyboard Connector Pins on CN5 36
VGA Video Connector CN18 37
Supported Video Resolutions and BIOS Settings 37
LVDS Flat Panel Video Connector CN19 38
EIDE Connector CN9 39
Bus Mouse Connector CN4 41
USB Connector CN17 42
10/100 Base T and TX Connector CN20 43
Audio Connector CN11 44
PC/104 XT Bus Connector, CN1 45
PC/104 AT Bus Connector, CN2 46
PC/104 Bus Signals 46
PC/104-Plus Bus Signal Assignments 49
SSD Devices 55
BIOS Setup Utility Keys 56
First Megabyte Memory Map 58
I/O Addresses Reserved for the cpuModule 59
Hardware Interrupts Used on the cpuModule 61
LED Colors 78
Manual LED Colors 78
Real Time Clock Registers 79
Jumpers and Default Settings 90
Solder Jumpers and Default Settings 90

CHAPTER 1: INTRODUCTION

This manual is meant for users developing with the CMX47786HX cpuModule. It contains information on hardware and software of the cpuModule.

READ THE SPECIFICATIONS FIRST.

The manual is organized as follows:

Chapter 1: Introduction

Introduces main features and specifications.

Chapter 2: Getting Started

Provides abbreviated instructions to get started.

Chapter 3: Connecting the cpuModule

Provides information on connecting the cpuModule to peripherals.

Chapter 4: Configuring the cpuModule

Provides information on configuring hardware and software.

Chapter 5: Using the cpuModule

Provides information needed to develop applications for the cpuModule. The chapter includes general information on the cpuModule, plus detailed information on storing applications and system functions, and using utility programs.

Chapter 6: Hardware Reference

Lists jumpers and their locations and mechanical dimensions.

Chapter 7: Troubleshooting

Offers advice on debugging problems with your system.

Chapter 8: Warranty

The CMX47786HX cpuModule

The PC/104 cpuModules described in this manual are designed for industrial applications which require:

- software and hardware compatibility with the PC/AT world
- high-speed “number-crunching” operation
- low power consumption
- small physical dimensions
- high reliability
- good noise immunity

This cpuModule is highly integrated, combining all major functions of a PC/AT computer on one compact board. It integrates all primary I/O functions of a AT compatible computer:

- SVGA controller
- Analog SVGA output
- 18 bit Digital LVDS Output
- UltraDMA 33/66/100 EIDE Controller
- a keyboard interface
- parallel port
- two versatile RS232/422/485 serial ports
- a Real Time Clock
- a speaker port
- a PS/2 mouse port
- two USB ports
- AC97 and Sound Blaster compatible audio port
- One twisted pair 10/100 Base T and TX connection based on an Intel 82559ER PHY

It also enhances standard AT-compatible computer systems by adding:

- Solid State Disk socket
- Non-volatile configuration without a battery
- Watchdog Timer
- Fail Safe Boot ROM
- A multiPort that can be configured as an ECP/EPP Parallel Port, 18 bit digital I/O, or Floppy Drive port

A simplified block diagram of the cpuModule is shown in Figure 1.

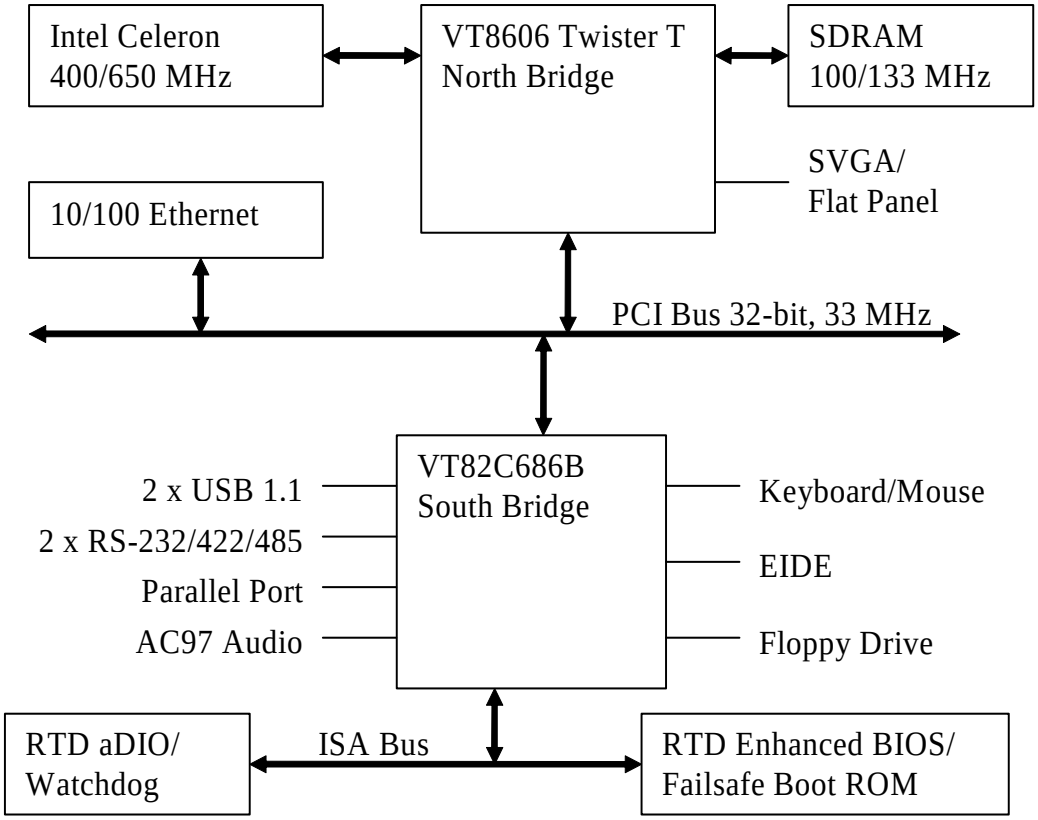


Figure 1: CMX47786HX Block Diagram

You can easily customize the cpuModule by stacking PC/104 modules such as video controllers, modems, LAN controllers, or analog and digital data acquisition modules. Stacking PC/104 modules on the cpuModule avoids expensive installations of backplanes and card cages and preserves the module's compactness.

RTD Enhanced Award BIOS is also implemented in the cpuModule. This BIOS supports ROM-DOS™, MS-DOS, Linux, and Windows operating systems. Drivers in the BIOS allow booting from floppy disk, hard disk, Solid State Disk, boot block flash, or DiskOnChip®, thus enabling the system to be used with traditional disk drives or non-mechanical drives.

The cpuModule and BIOS are also compatible with most real-time operating systems for PC compatible computers, although these may require creation of custom drivers to use the SSD and watchdog timer.

Specifications

CMX47786HX

- Intel Celeron CPU with Twister-T Chipset
- 400 MHz to 650 MHz clock speed.
- 0.95/1.10 V processor supply (provided on-board)
- 32 kB L1 Cache
- 256 kB L2 Cache
- Math coprocessor
- 100 MHz Front Side Bus

Video Controller

- AGP S3 Savage SVGA Controller
- Analog SVGA Output
- 18 bit LVDS Digital TFT Output
- 2 / 4 / 8 / 16 / 32 MB Frame Buffer

DMA, Interrupts, Timers

- Six (6) DMA channels (8237 compatible)
- Fifteen (15) interrupt channels (8259 compatible)
- Three (3) counter/timers (8254 compatible)
- Advanced Programmable Interrupt Controller (APIC)

USB ports

- 2 USB 1.1 ports
- Boot to USB devices
- Floppy
- Hard Drive
- CD-ROM
- Flash Key
- ZIP Drive

Advanced Digital I/O (aDIO)

- Two 8 bit, TTL compatible, programmable Digital I/O Ports plus two Strobe inputs.
- One port is bit direction programmable and the other is byte direction programmable.
- Advanced Interrupt modes
- Interrupt on change
- Interrupt on match
- Interrupt on strobe

Watchdog Timer

- Selectable reset time-out of 2sec, 1sec, 0.75 sec., 0.50 sec.
- Can generate an interrupt after 1sec, 0.75 sec., 0.50 sec., 0.25 sec.
- Watchdog Refresh and Active registers are accessible from Window

Memory Configurations

- 66/100/133 MHz SDRAM (selectable in BIOS)
- Available Memory Sizes:
- 32MB
- 128MB
- 256MB
- 512MB

Fail-safe Boot ROM

- Surface mount Flash chip that holds ROM-DOS™
- Provides an operating system with nothing else attached

Solid State Disk Socket

- SSD socket can hold one of the following Conventional SSD devices or BIOS Extension Devices.

Table 1: SSD Conventional Devices

Device	Full Read/Write Access	128Kbytes	256Kbytes	512Kbytes	1Mbyte
Atmel 5V Flash	yes	yes	yes	yes	no
SRAM	yes	yes	no	yes	no
NOVRAM	yes	yes	no	yes	no
EPROM	Read Only	yes	yes	yes	yes
AMD 5V Flash	Read Only	yes	no	yes	no

Table 2: SSD BIOS Extension Devices

Device	Full Read/Write Access	Maximum Number per cpuModule	Sizes
DiskOnChip® 2000	yes	1	16MB - 1GB *

(*) Larger devices may be available in the future.

Peripherals

- Two serial ports software configurable for RS232/422/485
- Baud rates from 50 to 115200 baud.
- Parallel port with SPP, ECP, EPP capability and selectable interrupts and DMA channel
- PC/AT standard keyboard port
- A PS/2 mouse port
- PC speaker port
- Real Time Clock (requires user-supplied external battery for date and time backup)

BIOS

- RTD Enhanced Award BIOS
- Directly supports Solid State Disk (SSD) and M-Systems' DiskOnChip®
- User-configurable using built-in Setup program
- Nonvolatile configuration without a battery
- Can boot from floppy disk, hard disk, Solid State Disk, fail-safe boot ROM, or USB.

Connections

- AT bus, per PC/104 specifications (64-pin CN1, 40-pin CN2)
- PCI bus, per PC/104-*Plus* specifications (120-pin CN16)
- Auxiliary Power Connector (12-pin CN3)
- PS/2 Mouse Connector (4-pin CN4)
- Multifunction connector (10-pin CN5)
- multiPort connector (26-pin CN6)

- Serial port 1 connector (10-pin CN7)
- Serial port 2 connector (10-pin CN8)
- SVGA Monitor connector (10-pin CN18)
- LVDS Flat Panel connector (20-pin CN19)
- Dual USB port connector(10-pin CN17)
- EIDE Hard Drive Connector (44-pin CN9)
- 10/100 Base T and TX connector(10-pin CN18)
- AC97 Audio input and output (10-pin CN11)
- CMOS Battery Connector (2-pin JP8)
- Fan Power Connectors (2-pin JP3 and JP7)

Physical Characteristics

- Weight (mass): 4.5 ounces (130 grams)
- PCB: 14-layer, mixed surface-mount and through-hole

Operating environment

- Power supply: 5V +/- 5%, 20 Watts
- Operating temperature: -40 to +80 degrees C case (with proper cooling) See ***Processor Thermal Management*** on page 105
- Storage temperature: -55 to +125 degrees C.
- Operating relative humidity: 0 to 95%, non-condensing

Power Consumption

Exact power consumption depends on the peripherals connected to the board, the selected SSD configuration and the memory configuration.

The table below lists power consumption for typical configurations and clock speeds:

Table 3: Power Consumption

Module	Speed	RAM	SSD	Consumption, Typical	Consumption Maximum
CMX47786HX	400 MHz	32 - 512 MB	None	7.6 W	11.6 W
CMX47786HX	650 MHz	32 - 512 MB	None	9.1 W	14.5 W

CHAPTER 2: GETTING STARTED

For many users, the factory configuration of the cpuModule can be used to get a PC/104 system operational. If you are one of these users, you can get your system up and running quickly by following a few simple steps described in this chapter. Briefly, these steps are:

- Connect power.
- Connect the utility cable.
- Connect a keyboard.
- Default BIOS Configuration
- Fail Safe Boot ROM
- Connect a VGA monitor to the SVGA connector.

Refer to the remainder of this chapter for details on each of these steps.

Basic Connector Locations

The following figure and table show the connectors used in this chapter

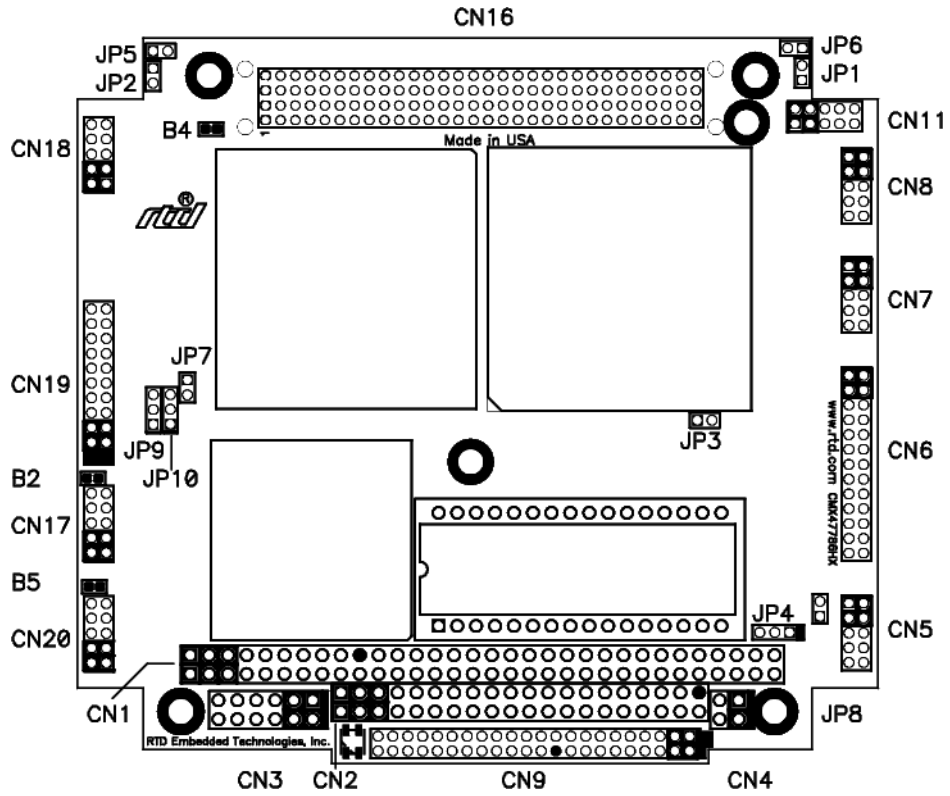


Figure 2: CMX47786HX Basic Connector Locations

For a complete listing of connectors, please refer to ***Connector Locations*** on page 24.

NOTE!	Pin 1 of each connector is indicated by a square solder pad on the bottom of the PC board and a white square silk-screened on the top of the board.

Fail safe boot ROM

Fail Safe Boot ROM is supplied with the board. This feature is programmed into a surface mount flash chip. The programmed boot ROM is ROM-DOS™. Fail Safe Boot ROM allows the system to boot without any attached storage devices. i.e. floppy, IDE, SSD. Installing jumper JP5 will force the cpuModule to use fail safe boot ROM, as well as load the default BIOS settings. This configuration allows you to boot to non-volatile onboard ROM-DOS™.

Cable Kits

For maximum flexibility, cables are not provided with the cpuModule. You may wish to purchase our cable kit for the cpuModule.

The XK-CM43 cable kit contains the following:

- Utility Board Cable (keyboard, mouse, battery, reset, speaker, SVGA, COM ports, Multi-Function Port)
- IDE Cable (for 3.5" Hard Drive)
- IDE Cable (for 2.5" Hard Drive)
- 10/100 base T and TX (10 Pin DIL TO RJ-45)
- Audio Adapter (10 Pin DIL to 3 mini-jack connectors)
- Power cable (DIL12 to wire leads)
- Dual USB cable

Connecting Power

WARNING!	If you improperly connect power, the module will almost certainly be *damaged* or *destroyed*. Such damage is not warranted! Please verify connections to the module *before* applying power.

Power is normally supplied to the cpuModule through the PC/104 bus, connectors CN1 and CN2. If you are placing the cpuModule onto a PC/104 stack that has a power supply, you do not need to make additional connections to supply power.

If you are using the cpuModule without a PC/104 stack or with a stack that does not include a power supply, refer to ***Auxiliary Power CN3*** on page 26 for more details.

Connecting the utility cable

The Multifunction connector, CN5, implements the following interfaces:

- AT keyboard
- Speaker output
- System reset input
- Battery input

To use these interfaces, you must connect to the Multifunction connector, making sure the orientation of pin 1 is correct. If you are using the Multifunction cable from our cable kit, the cable provides a small speaker, a 5-pin PS/2 connector for the keyboard, a push-button for resetting the PC/104 system, and a lithium battery to provide backup power to the Real Time Clock.

To connect individual devices to the Multifunction connector, please see ***Multifunction Connector, CN5*** on page 35**.**

Connecting a Keyboard

You may plug a PC/AT compatible keyboard directly into the circular DIN connector of the Multi-function cable in our cable kit. The cable kit uses a "mini-DIN," or PS/2 style keyboard connector.

NOTE!	Some older keyboards use a larger DIN connector; you will need an adapter to plug these keyboards into the cpuModule. Older keyboards can be switched between PC/XT and AT operating modes, with the mode usually selected by a switch on the back or bottom of the keyboard. For correct operation with this cpuModule, you must select AT mode.

Connecting to the PC/104 Bus

The PC/104 bus connectors of the cpuModule are simply plugged onto a PC/104 stack to connect to other devices.

We recommend you follow the procedure below to ensure that stacking of the modules does not damage connectors or electronics.

WARNING!	Do not force the module onto the stack! Wiggling the module or applying too much pressure may damage it. If the module does not readily press into place, remove it, check for bent pins or out-of-place keying pins, and try again.

- Turn off power to the PC/104 system or stack.
- Select and install standoffs to properly position the cpuModule on the PC/104 stack.
- Touch a grounded metal part of the rack to discharge any buildup of static electricity.
- Remove the cpuModule from its anti-static bag.
- Check that keying pins in the bus connector are properly positioned.
- Check the stacking order; make sure an XT bus card will not be placed between two AT bus cards or it will interrupt the AT bus signals.
- Hold the cpuModule by its edges and orient it so the bus connector pins line up with the matching connector on the stack.
- Gently and evenly press the cpuModule onto the PC/104 stack.

Connecting to the PC/104-Plus PCI Bus

The cpuModule is simply plugged onto a PC/104 stack. Other PC/104-*Plus* boards may then connect to the cpuModule's PC/104-*Plus* bus connector. Supplying power to the PCI bus is provided by the cpuModule.

We recommend you follow the procedure described for the PC/104 bus.

There are three additional considerations when using the PCI bus; PCI Bus signaling level, the slot selection switches on add in boards, and 3.3 volt power source for the expansion cards.

PCI Bus Signaling Levels

The PCI bus can operate at 3.3 or 5 volt signaling levels. This is controlled by solder blob jumper B1 and is configured at the factory for 3.3 volts from on board. If you desire to use 5 volt signaling, because you are connecting cards to the bus that require 5 volt signaling, you have to change the solder blob jumper B1. See ***Jumpers and Solder Jumper Settings*** on page 88 for details.

WARNING!	The bus can only operate as 3.3 Volt signaling **OR** 5 volt signaling, not both. You will have to ensure that all your expansion card can operate together at a single signaling level.

Slot Selection Switches

Unlike PC/104 cards, PC/104 *Plus* expansion cards have a "slot" selection switch or jumpers. In total, there are 4 PCI cards that can be stacked onto the cpuModule with switch positions 0 through 3. The distance from the CPU determines these switch settings. The card closest to the CPU is said to be in slot 0, the next closest slot 1 and so on to the final card as slot 3.

NOTE!	This requirement means that all PC/104 *Plus* cards must be stacked either on the top or the bottom of the CPU, not on both sides.

The "slot" setting method may vary from manufacturer to manufacturer, but the concept is the same. The CPU is designed to provide the correct delay to the clock signals to compensate for the bus length. The correct switch setting ensures the proper clock delay setting, interrupt assignment, and bus grant/request channel assignment. Refer to the expansion board's manual for the proper settings. Each expansion card must be in a different slot.

PCI Bus Expansion Card Power

+5 Volt DC

The +5 volt power pins on the PC/104 Plus PCI bus are directly connected to the +5 volt pins on the PC/104 connector and the power connector CN3 (pins 2 and 8). +5 volt expansion boards can be powered directly from these pins.

+3.3 Volt DC

The default source for the +3.3 volt power pins on the PC/104 Plus PCI bus is the power connector CN3, or a PC/104 Plus power supply. The +3.3V can also be supplied from an on board power converter. The on-board +5 volt to +3.3 volt converter is capable of suppling a maximum of 2 Amps of 3.3 volts to the PCI bus. To use the on-board power supply, change solder blob B3. See ***Jumpers and Solder Jumper Settings*** on page 88 for details.

Booting the cpuModule for the First Time

You can now apply power to the cpuModule. You will see a greeting message from the VGA BIOS and then:

- the cpuModule BIOS version information
- a message requesting you press {Del.} to enter the Setup program

If you don't press {Del.}, the cpuModule will try to boot from the current settings.

If you press {Del.}, the cpuModule will enter Setup. Once you have configured the cpuModule using Setup, save your changes and reboot.

NOTE!	By default, boards are shipped with fail safe boot ROM enable. When Fail Safe Boot ROM is enabled the system will boot to it exclusively.

Booting to Boot Block Flash with Fail Safe Boot ROM

The Fail Safe Boot ROM is a special build of ROM-DOS™ located inside a surface mounted Boot Block Flash chip that is memory mapped to the SSD window. Boot Block Flash is a write protected flash device that contains the BIOS and extra room where the Fail Safe Boot ROM is stored in the ROM DISK. The build is special because it can understand the ROM DISK format on the flash chip. Additionally, Fail Safe Boot ROM is an emergency interface accessible by an external computer. The ROM DISK contains REMDISK and REMSERVE for remote access to the system's disk drives. Due to the size of the flash chip, Fail Safe Boot ROM contains an abbreviated selection of the ROM-DOS™ utilities, however, the complete ROM-DOS™ is contained on a CD shipped with the board.

The purpose of the Fail Safe Boot ROM is to make the board bootable when the customer receives the cpuModule. Fail Safe Boot ROM can be used as an indicator of the board's functionality when booting problems arise with another operating system. This test can be accomplished by installing JP5. Installing JP5 forces the cpuModule to boot to Fail Safe Boot ROM, as well as load the default BIOS settings. The ROM DISK that contains the Fail Safe Boot ROM acts as an example of what can be programmed into the flash chip. Last, Fail Safe Boot ROM allows files to be transferred on or off the storage devices in the system by use of REMSERV and REMDISK, two ROM-DOS™ utilities.

If the user would need remote access to the system run REMSERV on the target system and REM-DISK on the host system. The end result would be that the storage devices on the target system would appear as additional drives on the host system. Information could then be transferred between hard disks by using a standard NULL Modem cable over a serial port. REMSERV makes the connection appear as an additional drive to the user. For details concerning this type of access, please refer to the ROM-DOS™ user's guide shipped with your board

NOTE!	By default, boards are shipped with fail safe boot ROM enable. When Fail Safe Boot ROM is enabled the system will boot to it exclusively.

The first time, your system will boot to the DOS prompt at the first available drive letter. If you do not intend to use REMSERV or REMDISK or you intend to boot from another device, you will need to disable Fail Safe Boot ROM. See the steps below for the method to disable it.

- Reset the system by either shutting it off and turning it on or by using the reset button.
- while the system is booting repeatedly press the DEL key to enter the BIOS setup.
- Choose INTEGRATED PERIPHERALS using the arrow keys and enter.
- Once in INTEGRATED PERIPHERALS set Fail Safe Boot ROM: Disabled

If You Misconfigure the cpuModule

It is possible that you may incorrectly configure the cpuModule using Setup. If this happens the correct procedure is:

- Start Re-booting the cpuModule.
- While the system is re-booting repeatedly press the {Del.} key until the cpuModule enters Setup.
- Change the parameters to correctly match your system.

If the above fails:

- Insert jumper JP5. This will force the cpuModule to boot using the default configuration.
- Boot the cpuModule.
- Press the {Del.} key to enter Setup, or allow the cpuModule to boot to Failsafe

For More Information

This chapter has been intended to get the typical user up and running quickly. If you need more details, please refer to the following chapters for more information on configuring and using the cpuModule.

Phone: 814-234-8087
Fax: 814-234-5218

RTD Embedded Technologies, Inc.
103 Innovation Blvd.
State College PA 16803-0906
USA

Our website: www.rtd.com
Our e-mail: techsupport@rtd.com

CHAPTER 3: CONNECTING THE CPUMODULE

This chapter contains necessary information for any of the connectors on the cpuModule.

Connector Locations

The figure and table below show all connectors and the SSD sockets of the cpuModule.

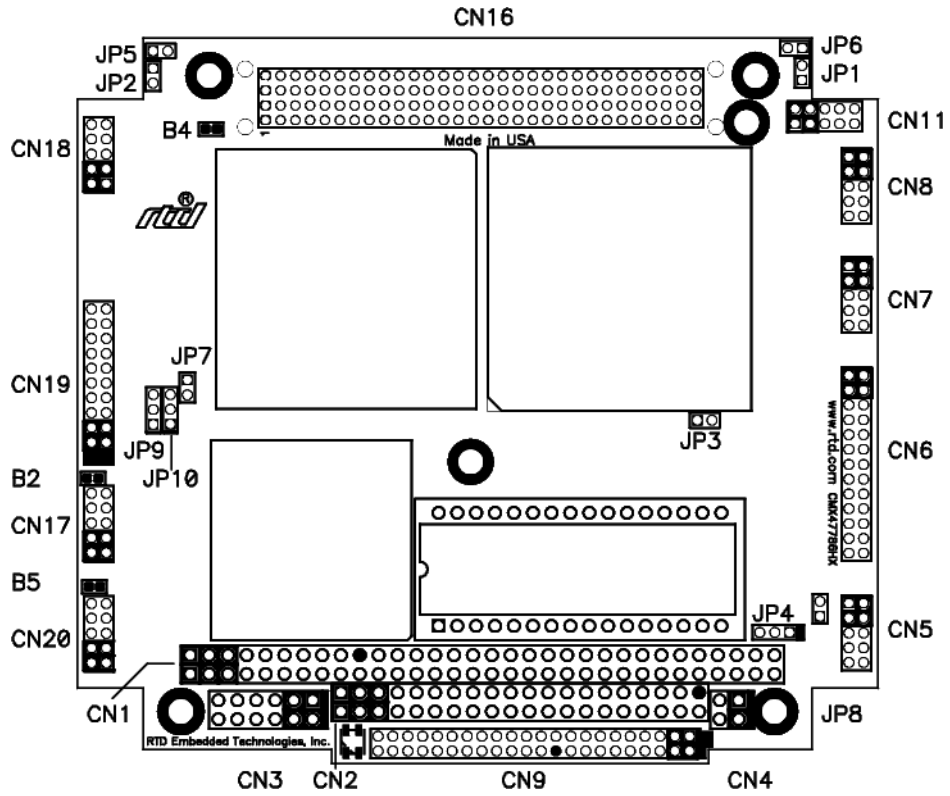


Figure 3: CMX47786HX Connector Locations

NOTE!	Pin 1 of each connector is indicated by a square solder pad on the bottom of the PC board and a white box silkscreened on the top of the board.

Table 4: CMX47786HX Connectors

Connector	Function	Size
CN1	PC/104 XT Bus	64 Pin, 0.1 inch
CN2	PC/104 AT Bus	40 Pin, 0.1 inch
CN3	Auxiliary Power	12 pin, 0.1 inch
CN4	Bus Mouse	4 pin, 0.1 inch
CN5	Multifunction	10 pin, 2mm
CN6	multiPort	26 pin, 2mm
CN7	Serial port 1	10 pin, 2mm
CN8	Serial port 2	10 pin, 2mm
CN9	EIDE Connector	44 pin, 2mm
CN11	Audio connector	10 pin, 2mm
CN16	PC/104-*Plus* PCI Bus	120 pin, 2mm
CN17	2 USB ports	10 pin, 2mm
CN18	Video	10 pin, 2mm
CN19	Flat Panel Video	30 pin, 2mm
CN20	10/100 Base T and TX	10 pin, 2mm
JP3	Switched Fan Power (1=5V, 2=GND)	2 Pin, 2mm
JP7	Continuous Fan Power (1=5V, 2=GND)	2 Pin, 2mm
JP8	RTC Battery (1=Vbatt, 2=GND)	2 Pin, 2mm

Auxiliary Power CN3

WARNING!	If you improperly connect power, the module will almost certainly be *destroyed*. Please verify power connections to the module *before* applying power.

The power supply can be conveyed to the module either through the PC/104 *Plus* bus (CN1, CN2, CN16) or through the Auxiliary Power Connector, CN3. The cpuModule only uses +5 VDC and ground. +12 VDC, -12 VDC and -5 VDC may be required on other PC/104 boards in the system.

Table 5: Auxiliary Power Connector CN3

Pin	Signal	Function
1	GND	Ground
2	+5 V	+5 Volts DC
3	N/C	Not Connected
4	+12 V	+12 Volts DC
5	-5 V	-5 Volts DC
6	-12 V	-12 Volts DC
7	GND	Ground
8	+5 V	+5 Volts DC
9	GND	Ground
10	+3.3 V	See Note
11	N/C	Not Connected
12	+3.3 V	See Note

NOTE!	Pins 10 and 12 on CN3 are connected to the +3.3V pins of the PC/104-*Plus* connector. They can be used to supply power to PC/104-*Plus* when a PC/104-*Plus* power supply is not being used.

Insufficient current supply will prevent your cpuModule from booting. The gauge and length of the wire used for connecting power to the cpuModule must be taken into consideration. Some power connectors have clip leads on them and may have significant resistance. Make sure that the input voltage does not drop below 4.8V at the 5V power pins. A good rule of thumb is to use wire that can supply twice the power your system requires.

NOTE!	-5 VDC, +12 VDC and -12 VDC voltages are not used by the module, but are connected to the PC/104 bus connectors CN1 and CN2.

Facing the connector pins, the pinout of the Auxiliary Power connector is:

11	9	7	5	3	1
N/C	GND	GND	-5V	N/C	GND
3.3 V	3.3 V	+5V	-12V	+12V	+5V
12	10	8	6	4	2

NOTE!	Connect two separate wires to the +5V pins (2 and 8) on the power connector to ensure a good power supply. We recommend that no less than 18 gauge wire be used and the length of this wire should not exceed 3 ft. Always measure the voltage drop from your power supply to the power pins on the cpuModule. The voltage at pins (2 and 8) should be +5V.

Serial Ports, CN7 and CN8

The two serial ports are implemented on connectors CN7 and CN8 respectively. Each port is normally configured as a PC compatible full-duplex RS232 port, but you may use the Setup program to re-configure it is as full-duplex RS422 or half-duplex RS485. The I/O address and corresponding interrupt must also be selected using Setup. The available I/O addresses and the corresponding interrupts are shown in the following table

Table 6: Serial Port Base Address and IRQ Options

I/O Address	IRQ
03F8H	IRQ4
02F8H	IRQ3
03E8H	IRQ4
02E8H	IRQ3

Serial Port UART

The serial port is implemented with a 16550-compatible UART (Universal Asynchronous Receiver/Transmitter). This UART is capable of baud rates up to 115.2 kbaud in 16450 and 16550A compatible mode, and includes a 16-byte FIFO. Please refer to any standard PC-AT hardware reference for the register map of the UART.

RS232 Serial Port (Default)

The full-duplex RS232 mode is the default setting on the cpuModule. With this mode enabled, the serial port connector must be connected to RS232 compatible devices. The following table gives the connector pinout and shows how to connect to an external serial connector, either DB25 or DB9 compatible.

Table 7: Serial Port in RS-232 Mode

Pin	Signal	Function	in/out	DB25	DB9
1	DCD	Data Carrier Detect	in	8	1
2	DSR	Data Set Ready	in	6	6
3	RXD	Receive Data	in	3	2
4	RTS	Request To Send	out	4	7
5	TXD	Transmit data	out	2	3
6	CTS	Clear To Send	in	5	8
7	DTR	Data Terminal Ready	out	20	4
8	RI	Ring Indicate	in	22	9
9,10	GND	Signal Ground	--	7	5

Facing the serial port's connector pins, the pinout is:

9	7	5	3	1
GND	DTR	TXD	RXD	DCD
GND	RI	CTS	RTS	DSR
10	8	6	4	2

RS422 or RS485 Serial Port

You may use BIOS Setup to configure the serial port as RS422 or RS485. In this case, you must connect the serial port to an RS422 or RS485 compatible device.

When using RS422 or RS485 mode, you can use the port in either half-duplex (two-wire) or full-duplex (four-wire) configurations. For half-duplex (2-wire) operation, you must connect RXD+ to TXD+, and connect RXD- to TXD-.

NOTE!	A 120 ohm termination resistors is provided on the cpuModule. Termination is usually necessary on all RS422 receivers and at the ends of the RS485 bus. If the termination resistor is required, it can be enabled by closing jumper JP1 for Serial Port 1, or JP2 for Serial Port 2.

When using full-duplex (typically in RS-422 mode) connect the ports as shown in the table below.

Table 8: RS-422 Serial Port Connections

Port 1	Port 2
RXD+	TXD+
TXD+	RXD+
RXD-	TXD-
TXD-	RXD-

When using half-duplex in RS-485 connect the ports as shown in the table below.

Table 9: RS-485 Serial Port Connections

From	To
Port 1 TXD+	Port 1 RXD+
Port 1 TXD-	Port 1 RXD-
Port 1 TXD+	Port 2 RXD+
Port 1 RXD-	Port 2 TXD-

RS422 and RS485 Mode Pinout

The following table gives the pinout of the serial port connector when RS422 or RS485 modes are enabled.

Table 10: Serial Port in RS-422/485 Mode

Pin	Signal	Function	in/out	DB9
1	--	Reserved	--	1
2	--	Reserved	--	6
3	RXD-	Receive Data (-)	in	2
4	TXD+	Transmit Data (+)	out	7
5	TXD-	Transmit Data (-)	out	3
6	RXD+	Receive Data (+)	in	8
7	--	Reserved	--	4
8	--	Reserved	--	9
9,10	GND	Signal ground	out	5

Facing the serial port connector, the pinout is:

9	7	5	3	1
GND	Rsvd	TXD-	RXD-	Rsvd
GND	Rsvd	RXD+	TXD+	Rsvd
10	8	6	4	2

NOTE!	***when using RS485 Mode*** When using the serial port in RS485 mode, the serial transmitters are enabled and disabled under software control. The transmitters are enabled by manipulating the Request To Send (RTS*) signal of the serial port controller. This signal is controlled by writing bit 1 of the Modem Control Register (MCR) as follows: • If MCR bit 1 = 1, then RTS* = 0, and serial transmitters are disabled • If MCR bit 1 = 0, then RTS* = 1, and serial transmitters are enabled For more information on the serial port registers, including the MCR, please refer to a standard PC-AT hardware reference for the 16550-type UART.

multiPort, CN6 - Parallel Port Mode

The parallel port is available on connector CN6. Make sure the BIOS setup sets the multiPort to parallel port. You can use the BIOS Setup to select the parallel port's address, associated interrupt, and choose between its operational modes (SPP, ECP, EPP 1.7 and EPP 1.9).

The pinout of the connector allows a ribbon cable to directly connect it to a DB25 connector, thus providing a standard PC compatible port.

The following tables lists the parallel port signals and explains how to connect it to a DB25 connector to obtain a PC compatible port.

NOTE!	For correct operation, keep the length of the cable connecting the cpuModule and parallel device less than 3 meters (10 feet).

Table 11: multiPort Parallel Port Connector Pinout CN6

Pin	Signal	Function	in/out	DB25
1	STB	Strobe Data	out	1
2	AFD	Autofeed	out	14
3	PD0	Printer Data 0 (LSB)	out	2
4	ERR	Printer Error	in	15
5	PD1	Parallel Data 1	out	3
6	INIT	Initialize printer	out	16
7	PD2	Printer Data 2	out	4
8	SLIN	Select printer	out	17
9	PD3	Printer Data 3	out	5
10	GND	Signal ground	--	18
11	PD4	Printer Data 4	out	6
12	GND	Signal ground	--	19
13	PD5	Printer Data 5	out	7
14	GND	Signal ground	--	20
15	PD6	Printer Data 6	out	8
16	GND	Signal ground	--	21
17	PD7	Printer Data 7 (MSB)	out	9
18	GND	Signal ground	--	22
19	ACK	Acknowledge	in	10
20	GND	Signal ground	--	23
21	BSY	Busy	in	11
22	GND	Signal ground	--	24
23	PE	Paper End	in	12
24	GND	Signal ground	--	25
25	SLCT	Ready To Receive	in	13
26	RSV	Reserved-Do Not Connect	--	--

multiPort, CN6 - aDIO Mode

Pin 1 is indicated by a square solder pad on the pin. This connector is located on the edge of the cpuModule along the PC/104 bus.

Table 12: multiPort Advanced Digital I/O Connector Pinout CN6

CN6 Pin	Function	DB25
1	strobe 0	1
2	P0-4	14
3	P1-0	2
4	P0-5	15
5	P1-1	3
6	P0-6	16
7	P1-2	4
8	P0-7	17
9	P1-3	5
10	strobe 1	18
11	P1-4	6
12	GND	19
13	P1-5	7
14	GND	20
15	P1-6	8
16	GND	21
17	P1-7	9
18	GND	22
19	P0-0	10
20	GND	23
21	P0-1	11
22	GND	24
23	P0-2	12
24	GND	25
25	P0-3	13
26	Reserved	Do not connect

multiPort, CN6 - Floppy Controller Mode

The MultiPort can be configured to be a floppy drive controller. This is selected in the BIOS Setup under Integrated Peripherals. Only one floppy drive can be connected to the MultiPort, and it is configured as the second drive.

NOTE!	To boot the CPU from the MultiPort Floppy, the following steps must be taken: • Drive A must be set to "1.44 MB" in the "Standard CMOS Settings" section of BIOS Setup. • Onboard MultiPort set to "Floppy" in the "Integrated Peripherals" section of BIOS Setup. • Swap Floppy Drive set to "Enabled" in the "Advanced BIOS Features" section of BIOS Setup. • First Boot Device set to "Floppy" in the "Advanced BIOS Features" section of BIOS Setup. • Floppy drive with adapter board is attached to CN6.

Pin 1 is indicated by a square solder pad on the pin. This connector is located on the edge of the cpuModule along the PC/104 bus. The following table shows the pin assignments to connect a floppy drive to the multiPort.

Table 13: multiPort Connector Floppy Pinout CN6

CN6 Pin	Function	DB25	Floppy Drive Pin
1	-	1	-
2	DR0	14	2
3	INDEX#	2	8 *
4	HDSEL#	15	32
5	TRK0#	3	26 *
6	DIR#	16	18
7	WRTPRT#	4	28 *
8	STEP#	17	20
9	RDATA#	5	30 *
10	GND	18	
11	DSKCHG	6	34 *
12	GND	19	odd pins
13	-	7	
14	GND	20	odd pins
15	-	8	
16	GND	21	odd pins
17	-	9	
18	GND	22	odd pins
19	DS1#	10	12
20	GND	23	odd pins
21	MTR#	11	16
22	GND	24	odd pins
23	WDATA#	12	22

Table 13: multiPort Connector Floppy Pinout CN6 (Continued)

24	GND	25	odd pins
25	WGATE#	13	24
26	+5V	-	

* These signals must be pulled to 5V with separate 470 Ohm resistors

Multifunction Connector, CN5

The Multifunction connector implements the following functions:

- Speaker output
- AT keyboard
- System reset input
- Battery Input

The following table gives the pinout of the Multifunction connector.

Table 14: Multifunction Connector CN5

Pin	Signal	Function	in/out
1	SPKR+	Speaker output (open collector)	out
2	SPKR-	Speaker output (+5 volts)	out
3	RESET#	Manual push button reset	in
4	PWRSW#	Power / Standby Switch	--
5	KBD	Keyboard Data	in
6	KBC	Keyboard Clock	out
7	GND	Ground	--
8	KBP	Keyboard Power (+5 volts)	out
9	BAT	Battery input	in
10	--	Not Connected	--

Facing the connector pins, the pinout is:

9	7	5	3	1
BAT	GND	KBD	RESET	SPKR+
	KBP	KBC		SPKR-
10	8	6	4	2

Speaker

A speaker output is available on pins 1 and 2 of the Multifunction connector. These outputs are controlled by a transistor to supply 0.1 watt of power to an external speaker. The external speaker should have 8 ohm impedance and be connected between pins 1 and 2.

Keyboard

An AT compatible keyboard can be connected to the Multifunction connector. Usually PC keyboards come with a cable ending with a 5-pin male PS/2 connector. The following table lists the relationship between the Multifunction connector pins and a standard PS/2 keyboard connector.

Table 15: Keyboard Connector Pins on CN5

Pin	Signal	Function	PS/2
5	KBD	Keyboard Data	1
6	KBC	Keyboard Clock	5
7	GND	Ground	3
8	KBP	Keyboard Power (+5 Volts)	4

System Reset

Pin 3 of the multifunction connector allows connection of an external push-button to manually reset the system. The push-button should be normally open, and connect to ground when pushed.

Power Switch

Pin 4 of the multifunction connector allows connection of an external push-button switch to act as an ATX power switch, or standby switch. The push-button should be normally open, and connect to ground when pushed.

Battery

Pin 9 of the multifunction connector is the connection for an external backup battery (in the range 2.40 V to 4.15 V; typically 3.0 or 3.6 V). This battery is used by the cpuModule when system power is removed, to preserve the date and time in the Real Time Clock and preserve SRAM contents if the SSD jumpers are configured for battery backup.

VGA Video Connector, CN18

The following table gives the pinout of the video connector.

Table 16: VGA Video Connector CN18

Pin	Signal	Function	in/out	DB15
1	VSYNC	Vertical Sync	out	14
2	HSYNC	Horizontal Sync	out	13
3	DDCSCL	Monitor communications clock	in/out	15
4	RED	Red analog output	out	1
5	DDCSDA	Monitor communications data	bidir	12
6	GREEN	Green analog output	out	2
7	DDC_5V	+5V for Monitor Communications (2A fuse)	out	nc
8	BLUE	Blue analog output	out	3
9	GND	Ground	out	5-7
10	GND	Ground	out	8,10

Facing the connector pins of CN18, the pinout is:

9	7	5	3	1
GND	DDC_5V	DDCSDA	DDCSCL	VSYNC
GND	BLUE	GREEN	RED	HSYNC
10	8	6	4	2

Table 17: Supported Video Resolutions and BIOS Settings

Resolution	Colors Bits	Minimum Frame Buffer Size
640 x 480	8/16/32	2 MB
800 x 600	8/16	2 MB
800 x 600	32	4 MB
1024 x 768	8/16	2 MB
1024 x 768	32	4 MB
1280 x 1024	8	2 MB
1280 x 1024	16	4 MB
1280 x 1024	32	8 MB
1600 x 1200	8/16	8 MB

LVDS Flat Panel Video Connector, CN19

The following table gives the pinout of the flat panel video connector. FP_VCC can be either 5V or 3V, and is selected with JP10. FP_VBKLT can be either 5V or 12V, and can be selected with JP9. See ***Jumpers and Solder Jumper Settings*** on page 88 for more details.

Table 18: LVDS Flat Panel Video Connector CN19

Pin	Signal	Function	in/out
1	Y0P	LVDS Data 0+	out
2	Y0M	LVDS Data 0-	out
3	GND	Ground	GND
4	GND	Ground	GND
5	Y1P	LVDS Data 1+	out
6	Y1M	LVDS Data 1-	out
7	GND	Ground	GND
8	GND	Ground	GND
9	Y2P	LVDS Data 2+	out
10	Y2M	LVDS Data 2-	out
11	GND	Ground	GND
12	GND	Ground	GND
13	YCP	LVDS Clock+	out
14	YCM	LVDS Clock-	out
15	GND	Ground	GND
16	GND	Ground	GND
17	GND	Ground	GND
18	FP_VCC	Power for flat panel electronics	out
19	FP_VBKLT	Power for flat panel backlight	out
20	FP_ENABLK	Enable for Backlight Power	out

EIDE Connector, CN9

The EIDE connector is a 44-pin, 2mm connector that can connect to a variety of EIDE or IDE devices. The connector provides all signals and power needed to use a 2.5" form factor (laptop) hard drive. Also, the first 40 pins of the connector provide all of the signals needed to interface to a 3.5"or 5" form factor hard drive, CD-ROM drive, or other EIDE device. The larger form factors use a 40-pin, 0.1 inch spacing connector, so an adapter cable or adapter board is needed.

In order to use faster than UDMA Mode 2 (Ultra ATA/33), an 80 conductor cable is required. The BIOS automatically detects the presence of an 80 conductor cable. To connect to an 80 conductor cable to CN12, and adapter board is required.

Table 19: EIDE Connector CN9

Pin	Signal
1	RESET-
2	Ground
3	DD7
4	DD8
5	DD6
6	DD9
7	DD5
8	DD10
9	DD4
10	DD11
11	DD3
12	DD12
13	DD2
14	DD13
15	DD1
16	DD14
17	DD0
18	DD15
19	Ground
20	N.C.)
21	DMARQ
22	Ground
23	DIOW-:STOP
24	Ground
25	DIOR-:HDMARDY-:HSTROBE
26	Ground
27	IORDY:DDMARDY-:DSTROB
28	CSEL
29	DMACK-
30	Ground
31	INTRQ
32	N.C
33	DA1
34	PDIAGDA0
35	N.C
36	DA2

Table 19: EIDE Connector CN9 (Continued)

37	CS0-
38	CS1-
39	DASP-
40	Ground
41	+5V(logic)
42	+5V(motor)
43	Ground
44	N.C.

Bus Mouse Connector, CN4

The following table gives the pinout of the Bus Mouse connector.

Table 20: Bus Mouse Connector CN4

Pin	Signal	Function	in/out
1	+5 V	+5 Volts	out
2	GND	Ground	out
3	MCLK	Mouse Clock	out
4	MDAT	Mouse Data	bidir

Facing the connector pins, the pinout is:

3	1
MCLK	+5 V
MDAT	GND
4	2

USB Connector,CN17

Two USB 1.1 compliant ports are available on CN17. The following table gives the pinout of the USB connector.

Table 21: USB Connector CN17

9 PIN D Pin	10 PIN DIL Pin	Signal	Function	in/out
1	**1**	**VCC1**	**Supply 5V to USB1**	**out**
6	**2**	**VCC2**	**Supply 5V to USB2**	**out**
2	**3**	**DATA1-**	**Bi-directional data line for USB1**	**in/out**
7	**4**	**DATA2-**	**Bi-directional data line for USB2**	**in/out**
3	**5**	**DATA1+**	**Bi-directional data line for USB1**	**in/out**
8	**6**	**DATA2+**	**Bi-directional data line for USB2**	**in/out**
4	**7**	**GRND**	**Signal Ground**	**out**
9	**8**	**GRND**	**Signal Ground**	**out**
5	**9**	**GRND**	**Signal Ground**	**out**
--	**10**	**GRND**	**Signal Ground**	**out**

Facing the connector pins, the pinout is

9	7	5	3	1
GRND	GRND	DATA1+	DATA1-	VCC1
GRND	GRND	DATA2+	DATA2-	VCC2
10	8	6	4	2

10/100 Base T and TX Connector, CN20

The functionality of this port is based on the INTEL 82559ER Fast Ethernet PCI controller. The following table gives the pinout of CN18. The Ethernet controller can be disabled in the BIOS Set-up.

Table 22: 10/100 Base T and TX Connector CN20

RJ45 Pin	10 PIN DIL Pin	Signal	Function	in/out
3	1	RX+	Receive +	In
6	2	RX-	Receive -	In
1	5	TX+	Transmit +	Out
2	6	TX-	Transmit -	Out
4	3	CT	Termination connected to pin 4	
5	4	CT	Termination connected to pin 3	
7	7	CT	Termination connected to pin 8	
8	8	CT	Termination connected to pin 7	
--	9	AGND	Ground	
--	10	AGND	Ground	

Facing the connector the pinout is depicted below

9	7	5	3	1
AGND	CT	TX+	CT	RX+
AGND	CT	TX-	CT	RX-
10	8	6	4	2

Audio, CN11

An audio port is available on CN11. This is an full featured, AC97 compliant port that also provides for Sound Blaster compatibility. It provides a mono microphone input, stereo line level input, and a stereo output that can be configured as line level or headphone level. The output is configured in the BIOS setup utility (see ***Configuring with the RTD Enhanced Award BIOS*** on page 56). When used as a headphone output, it will drive 32 Ohm speaker at 50mW.

The line input is a pseudo-differential input. A separate ground is provided to reference the left and right channel. This ground must be externally attached to signal ground. For best results, connect the line input ground to signal ground at the signal source. The will eliminate any common mode noise between the source and the cpuModule.

Table 23: Audio Connector CN11

10 PIN DIL Pin	Signal	Function	in/out
1	MIC_VREF	2.2V Supply to bias microphones. 5mA max.	pwr
2	MIC_IN	Microphone input. 1V RMS or 0.1V RMS.	in
3	GND	Signal GND	gnd
4	LINE_IN_LEFT	Line level input for left channel. 1V RMS nominal.	in/out
5	LINE_IN_GND	Signal GND input for line input. THIS MUST BE CONNECTED TO GND. For best results, connect to GND at the signal source	in
6	LINE_IN_RIGHT	Line level input for right channel. 1V RMS nominal.	in/out
7	GND	Signal Ground	out
8	OUTPUT_LEFT	Left channel output. Selectable as line level (1V RMS) or headphone.	out
9	GND	Signal Ground	out
10	OUTPUT RIGHT	Left channel output. Selectable as line level (1V RMS) or headphone.	out

Facing the connector pins, the pinout is

9	7	5	3	1
GND	GND	LINE_IN_GND	GND	MIC_VREF
OUT_R	OUT_L	LINE_IN_R	LINE_IN_L	MIC_IN
10	8	6	4	2

PC/104 Bus, CN1 and CN2

Connectors CN1 and CN2 carry signals of the PC/104 bus; these signals match definitions of the IEEE 1284 standard. The following tables list the pinouts of the PC/104 bus connectors.

The following table lists the signals of the XT portion of the PC/104 bus (see Notes below AT Bus table).

Table 24: PC/104 XT Bus Connector, CN1

Pin	Row A	Row B
1	N.C.	0V
2	SD7	RESETDRV
3	SD6	+5V
4	SD5	IRQ2
5	SD4	-5V
6	SD3	DRQ2
7	SD2	-12V
8	SD1	N.C.
9	SD0	+12V
10	IOCHRDY	(Keying pin)
11	AEN	SMEMW*
12	SA19	SMEMR*
13	SA18	IOW*
14	SA17	IOR*
15	SA16	DACK3*
16	SA15	DRQ3
17	SA14	DACK1*
18	SA13	DRQ1
19	SA12	REFRESH*
20	SA11	SYSCLK
21	SA10	IRQ7
22	SA9	IRQ6
23	SA8	IRQ5
24	SA7	IRQ4
25	SA6	IRQ3
26	SA5	DACK2*
27	SA4	TC
28	SA3	BALE
29	SA2	+5V
30	SA1	OSC
31	SA0	0V
32	0V	0V

The following table lists signals of the AT portion of the PC/104 bus.

Table 25: PC/104 AT Bus Connector, CN2

Pin	Row C	Row D
0	0V	0V
1	SBHE*	MEMCS16*
2	LA23	IOCS16*
3	LA22	IRQ10
4	LA21	IRQ11
5	LA20	IRQ12
6	LA19	IRQ15
7	LA18	IRQ14
8	LA17	DACK0*
9	MEMR*	DRQ0
10	MEMW*	DACK5*
11	SD8	DRQ5
12	SD9	DACK6*
13	SD10	DRQ6
14	SD11	DACK7*
15	SD12	DRQ7
16	SD13	+5V*
17	SD14	MASTER*
18	SD15	0V
19	(Keying pin)	0V

Notes:

•ISA bus refresh is not supported by this cpuModule.

•ISA Masters are not supported by this cpuModule

•Keying pin positions have the pin cut on the bottom of the board and the hole plugged in the connector to prevent misalignment of stacked modules. This is a feature of the PC/104 specification and should be implemented on all mating PC/104 modules.

•Signals marked with (*) are active-low.

•All bus lines can drive a maximum current of 4 mA at TTL voltage levels.

PC/104 Bus Signals

The following table contains brief descriptions of the PC/104 bus signals.

Table 26: PC/104 Bus Signals

Signal	I/O	Description
AEN	O	Address ENable: when this line is active (high), it means a DMA transfer is being performed, and therefore, the DMA controller has control over the data bus, the address bus, and the control lines.
BALE	O	Bus Address Latch Enable, active high. When active, it indicates that address lines SA0 to SA19 are valid.

Table 26: PC/104 Bus Signals

DACKx*	O	DMA ACKnowledge x=0-7, active low, used to acknowledge DMA requests.
DRQx	I	DMA Request x=0-7: these are asynchronous lines used by peripheral devices to request DMA service. They have increasing priority from DRQ0 up to DRQ7. A DMA request is performed by setting the DRQ line high and keeping it high until the corresponding DACK line is activated.
ENDXFR*	I/O	This is the only synchronous signal of the PC/104 bus and it is active low. It indicates that the current bus cycle must be performed with 0 wait states. It is used only for 16-bit boards.
IOCHCHK*	I	I/O Channel Check, active low, indicates an error condition that cannot be corrected.
IOCHRDY	I	I/O Channel Ready: this line, usually high (ready) is pulled to a low level by devices which need longer bus cycles.
IOCS16*	I	I/O Chip Select 16-bit: this line, active low, is controlled by devices mapped in the I/O address space. It indicates they have a 16-bit bus width.
IOR*	O	I/O Read, active low, indicates when the devices present on the bus can send their information on the data bus.
IOW*	O	I/O Write, active low. When active, it allows the peripheral devices to read data present on the data bus.
IRQx	I	Interrupt Request: x = 2 to 15, active on rising edge. IRQ15 has top priority; the other lines have decreasing priority starting from IRQ14 down to IRQ2. An interrupt request is performed by changing the level of the corresponding line from low to high and keeping it high until the microprocessor has recognized it.
KEY	N/A	These locations contain mechanical keying pins to help prevent incorrect connector insertion.
LA23-LA17	O	These signals select a 128kbyte window in the 16Mbyte address space available on the bus.
MASTER*	I	During a DMA cycle, this active-low signal, indicates that a resource on the bus is about to drive the data and address lines.
MEMCS16*	I	Memory Chip Select 16-bit: this line, active low, is controlled by devices mapped in the memory address space and indicates they have a 16-bit bus width.
MEMR*	I/O	This active-low signal indicates a memory read operation. Devices using this signal must decode the address on lines LA23-LA17 and SA19-SA0.
MEMW*	I/O	This active-low signal indicates a memory write operation. Devices using this signal must decode the address on lines LA23-LA17 and SA19-SA0.
OSC	O	OSCillator: clock with a 70 ns period and a 50% duty cycle. It is a 14.31818 MHz always presents.
REFRESH*	I	This cpuModule does not support refresh on the ISA bus. This pin is pulled high with a 4.7 K ohm resistor and may be driven by another card in the PC/104 stack. This line is active low and indicates that the current bus cycle is a DRAM refresh cycle. The refresh cycles are activated every 15 microseconds.

Table 26: PC/104 Bus Signals

RESETDRV	O	This line, active high, is used to reset the devices on the bus, at power-on or after a reset command.
SA0-19	O	Address bits 0 to 19: these lines are used to address the memory space and the I/O space. SA0 is the least significant bit while SA19 is the most significant bit.
SBHE*	O	This active-low signal indicates a transfer of the most significant data byte (SD15-SD8).
SD8-15	I/O	Data bits: these are the high-byte data bus lines. SD8 is the least significant bit; SD15 the most significant bit.
SD0-7	I/O	Data bits: these are the low-byte data bus lines. SD0 is the least significant bit; SD7 the most significant bit.
SMEMR*	O	Memory Read command, active low.
SMEMW*	O	Memory Write command, active low.
SYSCLK	O	System Clock, 8.0MHz with a 50% duty cycle. Only driven during external bus cycles.
TC	O	Terminal Count: this line is active high and indicates the conclusion of a DMA transfer.

PC/104 Bus Termination

Termination of PC/104 bus signals is not recommended since this cpuModule incorporates source termination on bus signals and may cause malfunctions of the cpuModule.

PC/104-Plus PCI Bus, CN16

Connector CN16 carries the signals of the PC/104-*Plus* PCI bus. These signals match definitions of the PCI Local Bus specification Revision 2.1. The following tables list the pinouts of the PC/104-*Plus* bus connector.

Table 27: PC/104-*Plus* Bus Signal Assignments

Pin	A	B	C	D
1	GND/5.0V KEY[1]	Reserved	+5V	AD00
2	VI/O	AD02	AD01	+5V
3	AD05	GND	AD04	AD03
4	C/BE0*	AD07	GND	AD06
5	GND	AD09	AD08	GND
6	AD11	VI/O	AD10	M66EN
7	AD14	AD13	GND	AD12
8	+3.3V	C/BE1*	AD15	+3.3V
9	SERR*	GND	SB0*	PAR
10	GND	PERR*	+3.3V	SDONE
11	STOP*	+3.3V	LOCK*	GND
12	+3.3V	TRDY*	GND	DEVSEL*
13	FRAME*	GND	IRDY*	+3.3V
14	GND	AD16	+3.3V	C/BE2*
15	AD18	+3.3V	AD17	GND
16	AD21	AD20	GND	AD19
17	+3.3V	AD23	AD22	+3.3V
18	IDSEL0	GND	IDSEL1	IDSEL2
19	AD24	C/BE3*	VI/O	IDSEL3
20	GND	AD26	AD25	GND
21	AD29	+5V	AD28	AD27
22	+5V	AD30	GND	AD31
23	REQ0*	GND	REQ1*	VI/O
24	GND	REQ2*	+5V	GNT0*
25	GNT1*	VI/O	GNT2*	GND
26	+5V	CLK0	GND	CLK1
27	CLK2	+5V	CLK3	GND
28	GND	INTD*	+5V	RST*
29	+12V	INTA*	INTB*	INTC*
30	-12V	REQ3*	GNT3*	GND/3.3V KEY[1]

Notes:

•The KEY pins can be used to guarantee proper module installation. Pin-A1 can be removed and the female side plugged for 5.0V I/O signals. Pin-D30 can be modified in the same manner for 3.3V I/O. It is recommended that both KEY pins (A1 and D30) be electrically connected to GND for shielding.

PC/104-*Plus* PCI Bus Signals

The following are brief descriptions of the PC/104-*Plus* PCI bus signals.

Address and Data

AD[31:00] -- Address and Data are multiplexed. A bus transaction consists of an address cycle followed by one or more data cycles.

C/BE[3:0]* -- Bus Command/Byte Enables are multiplexed. During the address cycle, the command is defined. During the Data cycle, they define the byte enables.

PAR -- Parity is even on AD[31:00] and C/BE[3:0]* and is required.

Interface Control Pins

FRAME* -- Frame is driven by the current master to indicate the start of a transaction and will remain active until the final data cycle.

TRDY* -- Target Ready indicates the selected devices ability to complete the current data cycle of the transaction. Both IRDY* and TRDY* must be asserted to terminate a data cycle.

IRDY* -- Initiator Ready indicates the master's ability to complete the current data cycle of the transaction.

STOP* -- Stop indicates the current selected device is requesting the master to stop the current transaction.

DEVSEL* -- Device Select is driven by the target device when its address is decoded.

IDSEL -- Initialization Device Select is used as a chip-select during configuration.

LOCK* -- Lock indicates an operation that may require multiple transactions to complete.

Error Reporting

PERR* -- Parity Error is for reporting data parity errors.

SERR* -- System Error is for reporting address parity errors.

Arbitration

REQ* -- Request indicates to the arbitrator that this device desires use of the bus.

GNT* -- Grant indicates to the requesting device that access has been granted.

System

CLK -- Clock provides timing for all transactions on the PCI bus.

RST* -- Reset is used to bring PCI-specific registers to a known state.

Interrupts

INTA* -- Interrupt A is used to request Interrupts.

INTB* -- Interrupt B is used to request Interrupts only for multi-function devices.

INTC* -- Interrupt C is used to request Interrupts only for multi-function devices.

INTD* -- Interrupt D is used to request Interrupts only for multi-function devices.

Power Supplies and VIO

+5V -- +5 volt supply connected to PC/104 bus and power connector +5V supplies.

+12V -- +12 volt supply connected to PC/104 bus and power connector +12V supplies.

-12V -- -12 volt supply connected to PC/104 bus and power connector -12V supplies.

+3.3V -- +3.3 volt supply is an on-board converter which can deliver up to 2 amps.

VIO -- This signal typically is the I/O power to the bus drivers on a PCI bus card. B1 selects +3.3 or +5 volts to indicate +3.3 or +5 volt signaling. The default is +3.3 volts. No other device except this board should drive the VIO pin.

CHAPTER 4: CONFIGURING THE CPUMODULE (BIOS SETUP)

This chapter contains information to configure the cpuModule.

Topics covered in this chapter include:

- Disabling Fail Safe Boot ROM
- Adding SSD Memory
- Configuring Using the Setup Program

Disabling Fail Safe Boot ROM

- Be sure that JP5 is not installed.
- Reset the system by either shutting it off and turning it on or by using the reset button.
- While the system is booting repeatedly press the DEL key to enter the BIOS setup.
- Choose INTEGRATED PERIPHERALS using the arrow keys and enter.
- Once in INTEGRATED PERIPHERALS set Fail Safe Boot ROM: Disabled

Installing SSD Memory

This section explains how to add SSD devices to the cpuModule. This procedure is only necessary when you wish to add or change Solid State Disk memory devices

You may wish to install SSD memory to use the cpuModule as a “diskless” stand-alone device.

Refer to ***Storing Applications On-board*** on page 81 for more information on various SSD device types. Solid State Disk memories are placed in SSD socket U16.

The following table lists possible configurations for the SSD socket:

Table 28: SSD Devices

SSD Support				
Type	Part	Operation	Capacity	Notes
Atmel 5V Flash	29C010A	read/write	128KB	
	29C040A	read/write	512KB	
BIOS Extension Devices	DiskOnChip Boot Block Flash	read/write	to 1 GB+	
NOVRAM	DS1645Y	read/write	128KB	
	DS1650Y	read/write	512KB	
SRAM	128KB	read/write	128KB	battery backup
	512KB	read/write	512KB	battery backup
12V Flash	28F010	read-only	128KB	read-only
	28F020	read-only	256KB	read-only
AMD 5V Flash	29F010	read-only	128KB	read-only
	29F040	read-only	512KB	read-only
EPROM	27C010	read-only	128KB	read-only
	27C020	read-only	256KB	read-only
	27C040	read-only	512KB	read-only
	27C080	read-only	1MB	read-only

Configuring with the RTD Enhanced Award BIOS

The cpuModule Setup program allows you to customize the cpuModule's configuration. Selections made in Setup are stored on the board and are read by the BIOS at power-on. The BIOS setup utility includes a help pane that describes each option. For more information about the BIOS options, please contact the factory.

Starting Setup

You can run Setup by:

- Re-boot the cpuModule, and repeatedly press the {Del.} key.

When you are finished with Setup, save your changes and exit. The system will automatically reboot.

Using the Setup Program

All displays in Setup consist of two areas. The left area lists the available selections. The right area displays help messages which you should always read. To see the default setting, press F1. If you need more information about the various BIOS settings and what they do, contact RTD's technical support.

Field Selection

You move between fields in Setup using the keys listed below.

Table 29: BIOS Setup Utility Keys

Key	Function
←↑↓→	move between fields
+, -, <PgUp>, <PgDn>	selects next/previous values in fields
<Enter>	Go to the submenu for the field, or list options of current field.
<Esc>	to previous menu then to exit menu

CHAPTER 5: USING THE CPUMODULE

This chapter provides information for users who wish to develop their own applications programs for the cpuModule.

This chapter includes information on the following topics:

- Memory map
- I/O Address map
- Interrupts
- Power On Self Tests (POSTs)
- System Functions (Watchdog Timer, Real Time Clock)
- Storing Applications in Solid State Disk
- Utility Programs

Memory Map

The ISA portion of the cpuModule addresses memory using 24 address lines. This allows a maximum of 2^{24} locations, or 16 Megabytes of memory.

The table below shows how memory in the first megabyte is allocated in the system.

Table 30: First Megabyte Memory Map

F0000-FFFFFh	BIOS ROM	256 KB BIOS in Flash EPROM, shadowed into DRAM during runtime.
E0000-EFFFFh	BIOS Decompression	Used during boot to decompress the BIOS. This should not be used for hardware devices, however, it is free for use after the system has booted.
D0000-FFFFh	Available	This area is always available for hardware devices
CC000-CFFFFh (typical)	Legacy USB	This area is used for USB keyboard and USB boot. It can be freed by disabling USB Keyboard, and by not attaching any USB storage devices. Alternately, the USB controller can be disabled. This area is always immediately after the video BIOS.
C0000-CBFFFh (typical)	Video BIOS	If an external video card is used, the size of this area may change.
B8000-BFFFFh	CGA Memory	Used for CGA video modes
B0000-B7FFFh	Mono Memory	Used for Monochrome video modes
A0000-AFFFFh	EGA/VGA Memory	Used for EGA/VGA video modes
9FFFFh-00502h		DOS reserved memory area
00501h-00400h		BIOS data area
003FFh-00000h		Interrupt vector area

Memory beyond the first megabyte can be accessed in real mode, by using EMS or a similar memory manager. See your OS or programming language references for information on memory managers.

Input/Output Address Map

As with all standard PC/104 boards, the Input/Output (I/O) space is addressed by 10 address lines (SA0-SA9). This allows 2^{10} or 1024 distinct I/O addresses. Any add-on modules you install must therefore use I/O addresses in the range 0-1023 (decimal) or 000-3FF (hex).

If you add any PC/104 modules or other peripherals to the system you must ensure they *do not* use reserved addresses listed below, or malfunctions will occur. The exception to this is if the resource has been released by the user.

The table below lists I/O addresses reserved for the cpuModule.

Table 31: I/O Addresses Reserved for the cpuModule

Address Range	Bytes	Device
000H-00FH	16	DMA Controller
010H-01FH	16	Reserved for CPU
020H-021H	2	Interrupt Controller #1
022H-02FH	13	Reserved
040H-043H	4	Timer
060H-064H	5	Keyboard Interface
070H-071H	2	Real Time Clock port
080H-08FH	16	DMA page register
0A0H-0A1H	2	Interrupt controller #2
0C0H-0DFH	32	DMA controller #2
0F0H-0FFH	16	Math co-processor
100H-101H	2	Video Initialization
1F0H-1FFH	16	Hard disk [1]
200H-201H	2	Reserved
238H-23BH	4	Bus Mouse[4]
2E8H-2EFH	8	Serial Port[2]
2F8H-2FFH	8	Serial port [2]
378H-37FH	8	Parallel port [3]
3BCH-3BFH	4	Parallel port [3]
3E8H-3EFH	8	Serial port [2]

Table 31: I/O Addresses Reserved for the cpuModule (Continued)

3F0H-3F7H	8	Floppy disk [1]
3F8H-3FFH	8	Serial port [2]
(Select in BIOS)	4	aDIO (Advanced Digital I/O)

[1] If a floppy or IDE controller is not connected to the system, the I/O addresses listed will not be occupied.

[2] Only one of the I/O addresses shown for a Serial port is active at any time. You can use Setup to select which one is active or to disable it entirely.

[3] Only one of the I/O addresses shown for the Parallel printer port is active at any time. You can use Setup to select which one is active or to disable it entirely.

[4]If a PS2 mouse is not connected to the system, the I/O addresses listed will not be occupied.

Hardware Interrupts

If you add any PC/104 modules or other peripherals to the system you must ensure they *do not* use interrupts needed by the cpuModule, or malfunctions will occur

The cpuModule supports the standard PC interrupts listed in Table 32 on page 61. Interrupts not in use by hardware on the cpuModule itself are listed as 'available'. These interrupts are also assigned by the BIOS to Plug and Play and PCI devices. Even if there are no PC/104-Plus modules installed, the interrupts will be assigned to on board peripherals, such as USB or parallel ports. The BIOS cannot automatically detect ISA bus (PC/104) cards that are using interrupts (including aDIO and the Watchdog timer), so it may inadvertently assign another resource to an ISA interrupt. To prevent this from happening, any interrupt that is being used for an ISA resource should be set to “Legacy ISA” in the BIOS Setup utility under PnP/PCI configuration.

Table 32: Hardware Interrupts Used on the cpuModule

Interrupt	Normal Use	Source
0	Timer 0	On-board ISA device
1	Keyboard	On-board ISA device
2	Cascade of IRQ 8-15	On-board ISA device
3	COM2	On-board ISA device
4	COM1	On-board ISA device
5	available	XT bus
6	Floppy[a]	XT bus
7	Printer	On-board ISA device
8	Real Time Clock	On-board ISA device
9	available, routed to IRQ 2	XT bus
10	available	AT bus
11	available	AT bus
12	Bus mouse[b]	On-board ISA device
14	primary IDE hard disk[c]	AT bus
15	available sometimes used as secondary IDE hard disk	AT bus

a. IRQ6 is available for use if no floppy disk is present in the system and floppy disk is disabled in Setup.

b. IRQ12 is available if there is no PS/2 mouse in the system, and the PS/2 mouse controller is disabled in the BIOS setup.

c. IRQ14 is available for use if no primary hard drive controller is present in the system and hard disk is disabled in Setup.

All PCI devices are capable of sharing an interrupt. The PC/104 specification also provides a means for ISA devices to share an interrupt. However, interrupt sharing on the ISA bus is not supported by all devices.

The RTD Enhanced Award BIOS

The RTD Enhanced Award BIOS (Basic Input/Output System) is software that interfaces hardware-specific features of the cpuModule to an operating system (OS). Physically, the BIOS software is stored in a Flash EPROM on the cpuModule. Functions of the BIOS are divided into two parts:

The first part of the BIOS is known as POST (Power-On Self-Test) software, and it is active from the time power is applied until an OS boots (begins execution). POST software performs a series of hardware tests, sets up the machine as defined in Setup, and begins the boot of the OS;

The second part of the BIOS is known as the CORE BIOS. It is the normal interface between cpuModule hardware and the operating system which is in control. It is active from the time the OS boots until the cpuModule is turned off. The CORE BIOS provides the system with a series of software interrupts to control various hardware devices.

For more information about the BIOS, please see ***Configuring the cpuModule (BIOS Setup)*** on page 53.

Thermal Throttling

The CMX47786HX cpuModule supports thermal throttling. Thermal throttling allows the CPU to continue operating at a lower clock rate when the CPU core exceeds a temperature limit. The power consumption and heat dissipation are therefore lowered, reducing the thermal stress on the CPU. The CPU will return to its normal speed when the temperature returns to 10C below the throttling threshold.

Thermal throttling is enabled in the BIOS Setup utility. There are three levels; 25%, 50%, and 75%, which are the percent of full speed that the CPU will operate at when throttled. For example, with throttling set to 75%, a 400 MHz CPU will operate at 300 MHz during excessive temperatures. When enabled, thermal throttling is started and stopped without any intervention from the operating system or application software.

Because thermal throttling starts before the internal temperature limit of the CPU is reached, throttling may occur within the operating temperature range of the cpuModule.

When thermal throttling is disabled, the CPU will operate at its maximum speed regardless of temperature.

Direct Hardware Control

Some of the cpuModule hardware is controlled directly without using BIOS routines. These include:

- Advanced Digital I/O (aDIO)
- Watchdog Timer
- Real Time Clock Control
- Parallel Port Control

The following sections describe use of these features.

Advanced Digital I/O Ports (aDIO)

This board supports 16 bits of TTL/CMOS compatible digital I/O (TTL signalling) plus two strobe inputs. Use the BIOS setup to set the multiPort into its aDIO mode. These I/O lines are grouped into two ports, port 0 and port 1. Port 0 is bit direction programmable and Port 1 is byte programmable. Port 0 supports RTD's two Advanced Digital Interrupt modes, ADI. The two modes are match and event. Match mode generates an interrupt when an eight bit pattern is received in parallel that matches the match mask register. The second ADI mode generates an interrupt when a change occurs on any bit. In either mode masking can be used to monitor selected lines.

When the CPU boots all digital I/O line are programmed as inputs. What this condition means is the digital I/O line's initial state is undetermined. If the digital I/O lines must power up to a known state an external 10 K Ohm resistor must be added to pull the line high or low. Additionally, when the CPU boots up interrupts 5, 7, 10, 11, and 12 are masked off.

Setup Register Descriptions

The aDIO has a Setup Register and four Runtime Registers. The Setup Register is set by the BIOS, and can be adjusted by entering the BIOS Setup Utility, and going to Integrated Peripherals. See ***Configuring with the RTD Enhanced Award BIOS*** on page 56 for more details. The Setup Register may also be read by the driver to determine the base address and interrupt of the aDIO.

aDIO/Watchdog Setup Register (I/O Port 0x1F)

aDIO/WDT_Setup

D7	D6 D4	D3	D2 D0
Reserved	aDIO_Base	Reserved	aDIO_Irq
r	r/w	r/w	r/w
0	0	set by BIOS	0

aDIO_Base[2:0] Selects the base address for aDIO and the Watchdog Timer based on the following table:

aDIO_Base[2:0]	aDIO Base I/O Address	Watchdog Timer Runtime Register I/O Address
000	0x450	0x454
001	0x440	0x444
010	0x410	0x414
011	0x400	0x404
100	0x350	0x354
101	0x340	0x344
110	0x310	0x314
111	0x300	0x304

aDIO_IRQ[2:0] Selects the Interrupt assigned to the aDIO based on the following table.

aDIO_IRQ[2:0]	Interrupt
000	Disabled.
001	IRQ5
010	IRQ7
011	IRQ10
100	IRQ11
101	IRQ12
110	Reserved
111	Reserved

Runtime Register Descriptions

The location of the Runtime registers for the digital I/O lines is determined by the value in aDIO/WDT_Setup[aDIO_Base]. These 8 bit registers are written to zero upon power up. The register map for the runtime region is shown below:

aDIO Runtime Registers

Offset	Register Name
0	Port 0 Data
1	Port 1 Data
2	Multi-Function
3	DIO-Control

Digital I/O Register Set

Port 0 Data I/O address aDIO_Base + 0

D7	D6	D5	D4	D3	D2	D1	D0
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0

Port 0 Data register is a read/write bit direction programmable register. A particular bit can be set to input or output. A read of an input bit returns the value of port 0. A read of an output bit returns the last value written to Port 0. A write to an output bit sends that value to port zero.

Port 1 data I/O address aDIO_Base + 1

D7	D6	D5	D4	D3	D2	D1	D0
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

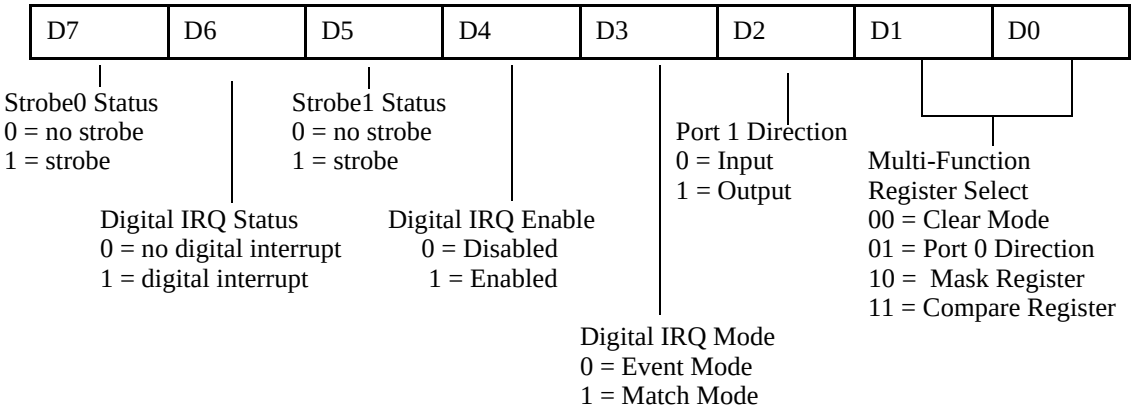
Port 1 Data register is a read/write byte direction programmable register. A read on this register when it is programmed to input will read the value at the DIO connector. A write on this register when it is programmed as output will write the value to the DIO connector. A read on this register when it is set to output will read the last value sent to the DIO connector.

Multi-Function I/O address aDIO_Base + 2

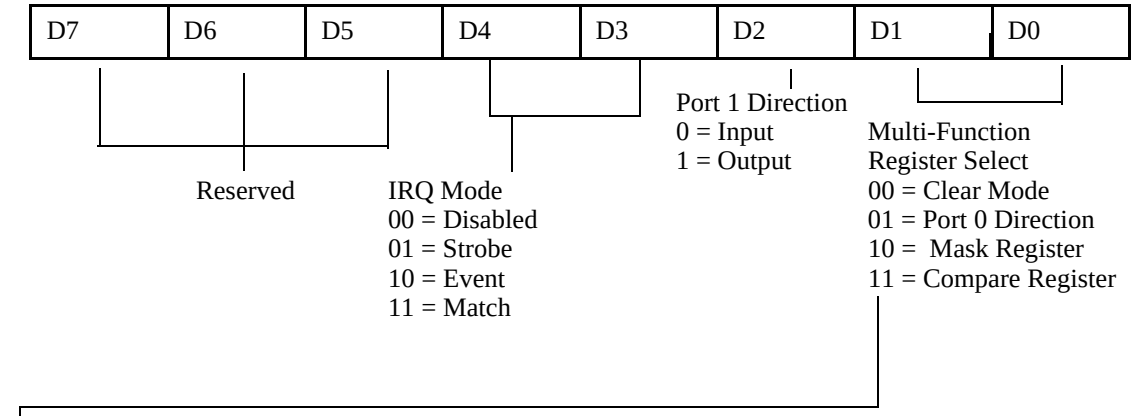
D7	D6	D5	D4	D3	D2	D1	D0

Multi-Function register is a read/write register whose contents are set by the DIO-Control register. See the DIO-Control register description for a description of this register.

DIO-Control I/O address aDIO_Base + 3 Read Access



DIO-Control I/O address aDIO_Base + 3 Write Access



Multi-Function, at aDIO_Base + 2, Contents Based on Bits D[1:0] of DIO-Control

read/write	**00 Clear**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**
0 = in/1 = out	**01 Port 0 Dir**	**I/O**	**I/O**	**I/O**	**I/O**	**I/O**	**I/O**	**I/O**	**I/O**
0 = no mask/1 = mask	**10 DIO Mask**	**M7**	**M6**	**M5**	**M4**	**M3**	**M2**	**M1**	**M0**
read/write	**11 Compare**	**C7**	**C6**	**C5**	**C4**	**C3**	**C2**	**C1**	**C0**

Clear Register:

A read to this register Clears the IRQs and a write to this register sets the DIO-Compare, DIO-Mask, DIO-Control, Port1 and Port0 to zeros. A write to this register is used to clear the board.

Port 0 Direction Register:

Writing a zero to a bit in this register makes the corresponding pin in the DIO connector an input. Writing a one to a bit in this register makes the corresponding pin in the DIO connector an output.

Mask Register:

Writing a zero to a bit in this register will not mask off the corresponding bit in the DIO-Compare register. Writing a one to a bit in this register masks off the corresponding bit in the DIO-Compare register. When all bits are masked off the DIOs comparator is disabled. This condition means Event and Match mode will not generate an interrupt. This register is used by Event and Match modes.

Compare Register:

A Read/Write register used for Match Mode. Bit values in this register that are not masked off are compared against the value on Port 0. A match or Event causes bit 6 of DIO-Control to be set and if the DIO is in Advanced interrupt mode, the Match or Event causes an interrupt.

Interrupts:

The Digital I/O can use interrupts 5, 7, 10, 11, and 12. The mapped interrupt numbers are 0x0D, 0x0F, 0x72, 0x73, and 0x74 in HEX respectfully or 13, 15, 114, 115, and 116 in decimal respectfully. To use any of the 5 listed interrupts set the interrupt aside for an ISA legacy device. To set the interrupts aside enter the BIOS under PNP/PCI CONFIGURATION. Select Resources Controlled By and change the interrupt(s) you wish to use to Legacy ISA. The interrupts you wish to use must then be selected in the Integrated Peripherals section of the BIOS under aDIO IRQ.

Advanced Digital Interrupts:

There are three advanced digital interrupt modes available. These three modes are Event, Match, and Strobe. The use of these three modes is to monitor state changes at the DIO connector. The three modes are selected with bits D[4:3] of the DIO-Control Register.

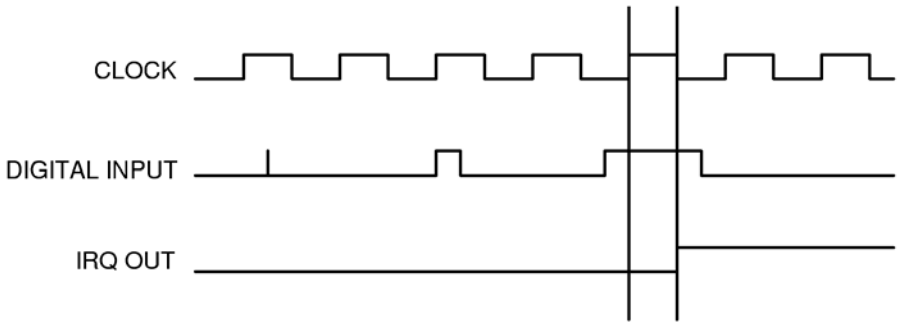
Event Mode:

When this mode is enable, Port 0 is latched into the DIO-Compare register at 8.33 MHz. There is a deglitching circuit inside the DIO circuitry. The deglitching requires pulses on Port 0 to be at least 120 nanoseconds in width. As long as changes are present longer than that, the event is guaranteed to register. Pulses as small as 60 nanoseconds can register as an event but they must occur between the rising and falling edge of the 8.33 MHz clock. To enter Event mode, set bits D[4:3] of the DIO-Control register to a “10”.

Match Mode:

When this mode is enabled, Port 0 is latched into the DIO-Compare register at 8.33 MHz. There is a deglitching circuit inside the DIO circuitry. The deglitching requires pulses on Port 0 to be at least 120 nanoseconds in width. As long as changes are present longer than that, the match is guaranteed to register. Pulses as small as 60 nanoseconds can register as a match but they must occur between the rising and falling edge of the 8.33 MHz clock. To enter Match mode, set bits D[4:3] of the DIO-Control register to “11”.

	Make sure bit 3 is set BEFORE writing the DIO-Compare register. If you do not set bit 3 first, the contents of the DIO-Compare register could be lost. The reason for this is that Event mode latches in Port 0 into the DIO-Compare register at an 8.33 MHz rate.



Strobe Mode:

Strobe Mode allows the strobe pin of the DIO connector to trigger an interrupt. A low to high transition on the strobe pin will cause an interrupt request. The request will remain high until the Clear Register is read from. Additionally, the Compare Register latched in the value at Port 0 when the Strobe pin made a low to high transition. No further strobes will be available until a read of the Compare Register is made. What this implies is one must read the Compare Register then clear interrupts so that the latched value in the compare register is not lost. To enter Strobe mode, set bits D[4:3] of the DIO-Control register to “01”.

Basic Interrupt Information for Programmers:

All information below only addresses the DIO on this board. Interrupts are connected to IRQs 5, 7, 10, 11, and 12 on the ISA bus (PC104 bus) and are controlled by two 8259-equivalent interrupt controllers containing 13 available interrupt request lines. Minimum time between two IRQ requests is 125 nanoseconds as set by ISA specification.

What is an Interrupt?

An interrupt is a subroutine called asynchronously by external hardware (usually an I/O device) during the execution of another application. The CPU halts execution of its current process by saving the system state and next instruction then jumps to the interrupt service routine, executes it, loads the saved system state and saved next instruction, and continues execution. Interrupts are good for handling infrequent events such as keyboard activity.

What happens when an Interrupt occurs?

An IRQx pin on the PC104 bus makes a low to high transition while the corresponding interrupt mask bit is unmasked and the PIC determines that the IRQ has priority, the PIC interrupts the processor. The current code segment (CS), instruction pointer (IP), and flags are pushed on the stack, the CPU reads the 8 bit vector number from the PIC and a new CS and IP are loaded from a vector, indicated by the vector number, from the interrupt vector table that exists in the lowest 1024 bytes of memory. The processor then begins executing instructions located at CS:IP. When the interrupt service routine is completed the CS, IP, and flags that were pushed onto the stack are popped from the stack into their appropriate registers and execution resumes from the point where it was interrupted.

How long does it take to respond to an interrupt?

A DOS operating system can respond to an interrupt between (6-15uS). A Windows system can take a much longer time when a service routine has been installed by a device driver implemented as a DLL; from 250-1500uS or longer. A VxD will take 20-60uS or longer. The time the CPU spends in the interrupt is dependent on the efficiency of the code in the ISR. These numbers are general guidelines and will fluctuate depending on operating system and version. The amount of information that can be moved during an interrupt theoretically can be 4 MB/Sec. on a 8 MB bus using the INS or MOVS instruction with the REP prefix. These instructions are in assembly language.

Interrupt Request Lines:

To allow different peripheral devices to generate interrupts on the same computer, the PC bus has eight different interrupt request (IRQ) lines. A transition from low to high on one of these lines generates an interrupt request which is handled by the PC's interrupt controller. The interrupt controller checks to see if interrupts are to be acknowledged from that IRQ and, if another interrupt is already in progress, it decides if the new request should supersede the one in progress or if it has to wait until the one in progress is done. This prioritizing allows an interrupt to be interrupted if the second request has a higher priority. The priority level is based on the number of the IRQ; IRQ0 has the highest priority, IRQ1 is second-highest, and so on through IRQ7, which has the lowest. Many of the IRQs are used by the standard system resources. IRQ0 is used by the system timer, IRQ1 is used by the keyboard, IRQ3 by COM2, IRQ4 by COM1, and IRQ6 by the disk drives. Therefore, it is important for you to know which IRQ lines are available in your system for use by the module.

8259 Programmable Interrupt Controller:

The chip responsible for handling interrupt requests in the PC is the 8259 Programmable Interrupt Controller. To use interrupts, you need to know how to read and set the 8259's interrupt mask register (IMR) and how to send the end-of-interrupt (EOI) command to the 8259.

Interrupt Mask Register (IMR):

Each bit in the interrupt mask register (IMR) contains the mask status of an IRQ line; bit 0 is for IRQ0, bit 1 is for IRQ1, and so on. If a bit is set (equal to 1), then the corresponding IRQ is masked and it will not generate an interrupt. If a bit is clear (equal to 0), then the corresponding IRQ is unmasked and can generate interrupts. The IMR is programmed through port 21H.

Writing an Interrupt Service Routine:

The first step in adding interrupts to your software is to write the interrupt service routine (ISR). This is the routine that will automatically be executed each time an interrupt request occurs on the specified IRQ. An ISR is different than standard routines that you write. First, on entrance, the processor registers should be pushed onto the stack BEFORE you do anything else. Second, just before exiting your ISR, you must clear the interrupt status flag of the DM5812 and write an end-of-interrupt command to the 8259 controller. Finally, when exiting the ISR, in addition to popping all the registers you pushed on entrance, you must use the IRET instruction and not a plain RET. The IRET automatically pops the flags, CS, and IP that were pushed when the interrupt was called.

If you find yourself intimidated by interrupt programming, take heart. Most C compilers allow you to identify a procedure (function) as an interrupt type and will automatically add these instructions to your ISR, with one important exception: most compilers do not automatically add the end-of-interrupt command to the procedure; you must do this yourself. Other than this and the few exceptions discussed below, you can write your ISR just like any other routine. It can call other functions and procedures in your program and it can access global data. If you are writing your first ISR, we recommend that you stick to the basics; just something that will convince you that it works, such as incrementing a global variable.

NOTE: If you are writing an ISR using assembly language, you are responsible for pushing and popping registers and using IRET instead of RET.

Writing a DOS Interrupt service routine (ISR):

There are a few cautions you must consider when writing your ISR. The most important is, do not use any DOS functions or routines that call DOS functions from within an ISR. DOS is not reentrant; that is, a DOS function cannot call itself. In typical programming, this will not happen because of the way DOS is written. But what about when using interrupts? Then, you could have a situation such as this in your program. If DOS function X is being executed when an interrupt occurs and the interrupt routine makes a call to DOS function X, then function X is essentially being called while it is already active. Such a reentrance attempt spells disaster because DOS functions are not written to support it. This is a complex concept and you do not need to understand it. Just make sure that you do not call any DOS functions from within your ISR. The one wrinkle is that, unfortunately, it is not obvious which library routines included with your compiler use DOS functions. A rule of thumb is that routines which write to the screen, or check the status of or read the keyboard, and any disk I/O routines use DOS and should be avoided in your ISR.

The same problem of reentrance exists for many floating point emulators as well, meaning you may have to avoid floating point math in your ISR.

The Code:

Refer to the DOS drivers that were shipped with this board or download them from our web site www.rtd.com. The drivers are commented to help clarify their meaning. Reading through the DOS drivers will give valuable insight into the board functionality.

Watchdog Timer Control

The cpuModule includes a Watchdog Timer, which provides protection against programs "hanging", or getting stuck in an execution loop where they cannot respond correctly. The watchdog timer consists of a counter, a reset generator, and an interrupt generator. When the counter reaches the interrupt time-out, it can generate an interrupt. When the counter reaches the reset time-out, the system is reset. The counter is "refreshed," or set back to zero by reading from a specific register. The watchdog can also be put into an "inactive" state, in which no resets or interrupts are generated.

The ability to generate an interrupt allows the application to gracefully recover from a bad state. For example, consider a system that has a reset time-out of 2 seconds, interrupt time-out of 1 second, and the watchdog timer is refreshed every 0.5 seconds. If something goes wrong, an interrupt is generated. The Interrupt service routine then attempts to restart the application software. If it is successful, the application is restarted in much less time than a full reboot would require. If it is not successful, the system is rebooted.

Due to system latency, it is recommended that the Watchdog be refreshed at about half of the reset time-out period, or half of the interrupt time-out period, whichever is applicable.

Register Description

The Advanced Watchdog Timer has two Setup Registers and a Runtime Register. The Setup Registers are set by the BIOS, and can be adjusted by entering the BIOS Setup Utility, and going to Integrated Peripherals. See ***Configuring with the RTD Enhanced Award BIOS*** on page 56 for more details. The Setup Register may also be read by the driver to determine if the Watchdog is enabled, and the interrupt and base address that it is using.

In the following register description sections, each register is described by a register table. The first row of the table list the bits, D7 through D0. The second row lists the field name for each bit. The third row lists the properties of that bit; 'r' = bit can be read, 'w' = bit can be written to, and 'c' = bit can be cleared. The last row lists the value of the bit after reset. The register table is then followed by a description of each of the fields where applicable.

Advanced Watchdog Setup Register (I/O Port 0x18)

WDT_Setup

D7	D6	D5	D4	D3 D1	D0
Reserved	Reserved	Reserved	Reserved	WDT_IRQ	Reg_Enable
r	r	r	r	r/w	r/w
0	0	0	0	0	0

WDT_IRQ[2:0] Selects the Interrupt assigned to the Watchdog Timer.

WDT_IRQ[2:0]	Interrupt
000	DIsabled.
001	IRQ5
010	IRQ7
011	IRQ10
100	IRQ11

(Continued)

WDT_IRQ[2:0]	Interrupt
101	IRQ12
110	Reserved
111	Reserved

Reg_Enable Set this bit to '1' to enable the Watchdog Runtime Register. Set to '0' to disable. When Disabled, the Watchdog Runtime Register does not appear in the I/O map, and interrupts and resets are not generated by the Watchdog Timer.

aDIO/Watchdog Setup Register (I/O Port 0x1F)

aDIO/WDT_Setup

D7	D6 D4	D3	D2 D0
Reserved	aDIO_Base	Reserved	aDIO_Irq
r	r/w	r/w	r/w
0	0	set by BIOS	0

aDIO_Base[2:0] Selects the base address for aDIO and the Watchdog Timer based on the following table:

aDIO_Base[2:0]	aDIO Base I/O Address	Watchdog Timer Runtime Register I/O Address
000	0x450	0x454
001	0x440	0x444
010	0x410	0x414
011	0x400	0x404
100	0x350	0x354
101	0x340	0x344
110	0x310	0x314
111	0x300	0x304

Watchdog Runtime Register

The location of this register is determined by aDIO/WDT_Setup[aDIO_Base]. Reading this register also refreshes the watchdog timer.

WDT_Runtime

D7	D6	D5	D4	D3 D2	D1 D0
WDT_Active	WDT_IRQ_Ena	Reserved	Reserved	WDT_IRQ_Ena	WDT_RST_Time
r/w	r/w	r	r	r/w	r/w
0	0	0	0	0	0

WDT_Active Set this bit to '1' to activate the Watchdog Timer interrupts and resets. Set to '0' to disable. When disabled, the watchdog timer does not need to be refreshed.

WDT_IRQ_Ena Set this bit to '1' to enable the Watchdog interrupt. Set to '0' to disable.

WDT_IRQ_TimeSelects the Watchdog Interrupt time based on the following table:.

WDT_IRQ_Time[2:0]	Interrupt Time (seconds)
00	1.00
01	0.75
10	0.50
11	0.25

WDT_RST_TimeSelects the Watchdog Reset time based on the following table:.

WDT_RST_Time[2:0]	Reset Time (seconds)
00	1.00
01	0.75
10	0.50
11	1.55-2.10

Multi-Color LED

The CMX47786HX has a Multi-Color LED located beside the IDE connector, CN9. The color of the LED indicates the status of the board, as shown in the table below:

Table 33: LED Colors

Color	Description
Green	Normal Operation
Blue	On Board IDE Activity
Red	cpuModule is in reset, or approaching thermal limit
Yellow (Red + Green)	cpuModule is in Standby
White (R+G+B)	cpuModule is approaching thermal limit (CPU is throttled if enabled)
Cyan (Blue + Green)	Ethernet Link at 10 Mbps
Magenta (Blue + Red)	Ethernet Link at 100 Mbps
Blink	Ethernet Activity

The LED can also be controlled manually by writing to I/O Port 0x16. The following table lists the color displayed and the value written.

Table 34: Manual LED Colors

I/O Port 0x16 Value	Color
0x00	Automatic (see ***LED Colors*** on page 78)
0x08	Off
0x09	Blue
0x0A	Green
0x0B	Cyan (Green + Blue)
0x0C	Red
0x0D	Magenta (Red + Blue)
0x0E	Yellow (Red + Green)
0x0F	White (Red + Green + Blue)

Real Time Clock Control

The cpuModule is equipped with a Real Time Clock (RTC) which provides system date and time functions, and also provides 128 non-volatile memory locations. The contents of these memory locations are retained whenever an external backup battery is connected, whether or not system power is connected.

You may access the RTC date, time, and memory using an index and data register at I/O addresses 70h and 71h. Address 70h is the Index register. It must be written with the number of the register to read or write. Refer to the map below for valid choices for the index. Data is then written to or read from the selected register by writing or reading (respectively) the data register at address 71h.

Do *not* change values stored in the RTC registers listed as RESERVED in the table below. Doing so will interfere with proper cpuModule operation.

Registers of the Real Time Clock are shown below:.

Table 35: Real Time Clock Registers

Registers (hex)	Registers (decimal)	Number of Bytes	Function
00h	0	1	BCD Seconds
02h	2	1	BCD Minutes
04h	4	1	BCD Hours
06h	6	1	Day of week
07h	7	1	Day of month
08h	8	1	Month
09h	9	1	Year
0A-31h	10-49	40	RESERVED- Do not modify!
32h	50	1	BCD Century
33-3Fh	51-63	13	RESERVED - Do not modify!
40-7Fh	64-127	64	User RAM

RTC access is illustrated in the C program found in application note SWM-640000011 found at http://www.rtd.com/appnotes.htm

Parallel Port Control

The parallel port may be operated in SPP (output-only), EPP (bi-directional), and ECP (extended capabilities) modes. The mode may be selected in Setup, or by application software.

Storing Applications On-board

The cpuModule was designed to be used in embedded computing applications. In these applications, magnetic media like hard disks and floppy disks are not very desirable. It is better to eliminate magnetic storage devices and place your operating system and application software into the cpuModule's Solid State Disk (SSD).

The following section describes two distinctly different ways you may use the Solid State Disk sockets of the cpuModule. These methods allows you to use a wide variety of memory devices to implement on-board Solid State Disk storage, each with its advantages and disadvantages.

Ways to Use the Solid State Disk Sockets

The ways to utilize the Solid State Disk sockets of the cpuModule.

- Using a device which installs as a BIOS Extension
- Using Conventional SSD Memory

BIOS Extension Devices such as DiskOnChip® provide a relatively large amount of read/write disk space. These devices generally appear similar to a conventional hard disk to DOS, allowing you to copy, delete, and rename files without using any special utilities.

Conventional Solid State Disk can use a variety of memory devices, such as:

- Atmel 5 volt only Flash
- Intel 12 volt Flash
- AMD 5-volt Flash
- Static RAM
- NOVRAM
- EPROM

When used with Atmel 5 volt Flash, SRAM, or NOVRAM, the SSD appears similar to a read/write floppy disk to DOS, allowing you to copy, delete, and rename files in the SSD without using any special utilities.

When used with other Flash or EPROM, you can create file to program the device with the RIMAGE program.

These methods are described in detail in the following sections.

Using BIOS Extension Devices

You can use BIOS Extension Devices like M-Systems DiskOnChip® to implement a Solid State Disk which can be read and written using normal disk commands.

Advantages of using these devices include:

- Storage capacity up to 1 GB per socket (more in the future)
- Full read/write capability using standard OS file commands
- Integrated support for other operating systems is possible (contact M-Systems for information)

Parts supported by the cpuModule include:

- M-Systems 2000 series: MD2203-D1024 (1 GB) down to...16 MB
- Optional alternate operating system versions of the above parts

Our website at www.rtd.com provides links to the websites of these manufacturers.

Installing BIOS Extension Devices

To install these devices, follow this procedure:

- Apply power to the cpuModule and run Setup.
- Set a BIOS Extension Window on the advanced setup page.

NOTE!	The memory window selected for a BIOS extension device must not be used by any other program or hardware device. Make sure this window is not used by EMM386 or another memory manager, Ethernet card, PCMCIA card, etc.

- Save your changes and exit Setup.
- Turn off the cpuModule.
- Install the BIOS Extension Device into the socket.
- Reboot the cpuModule. The BIOS Extension Device should appear as the next available hard drive in your system. If there is no other hard drive installed, it will appear as drive C:.
- Format the new drive using the DOS format command.

NOTE!	If you wish to make the drive bootable, you must format it using the **/s** switch of the format command. Refer to your OS manual for more information.

- If you wish to boot from the BIOS Extension Device, run Setup and disable any other hard drive. Set the boot device to Hard Drive. The cpuModule will not boot to a BIOS Extension Device if another hard drive is enabled.

Using Conventional Solid State Devices

You can use numerous memory types to implement a Conventional Solid State Disk. Depending on the devices used, you may implement read/write, read-only, or write-once-read-many type drives.

Advantages of using the Conventional SSD include:

- Storage capacity up to 1 MB (EPROM only)
- Atmel Flash and NVRAM allow read/write capability using standard DOS file commands

Disadvantages of using the Conventional SSD include:

- Requires external utility program and device programmer to program 12 volt Flash, AMD 5-volt Flash, or EPROM.
- Limited memory size.

The following memory devices or their equivalents may be used for a Conventional SSD. Access times for all devices must be 150 ns or less, and all devices *must* be in 32-pin DIP packages.

- Atmel 29C010A, 29C020A or 29C040A
- +12 V Flash 28F010, 28F020
- +5 V Flash 29F010, 29F040
- SRAM 128Kx8, 512Kx8
- NOVRAM (Dallas Semiconductor DS1645Y, DS1650Y)
- EPROM (27C010, 27C020, 27C040, 27C080)

Installing a Conventional SSD using Atmel 5-volt-only Flash

To install an SSD using Atmel Flash, follow this procedure:

- Apply power to the cpuModule and run Setup.
- Set **SSD socket 1 or 2** to the appropriate Atmel device type.
- Set **SSD Window** to a value which will not conflict with other hardware or software.

NOTE!	The memory window selected for DOC must not be used by any other program or hardware device. Make sure this window is not in use by EMM386 or another memory manager, or an Ethernet card, PCMCIA card, etc.

- Save your changes and exit Setup.
- Turn off the cpuModule.
- Install the memory device into the socket.
- Reboot the cpuModule. The SSD should appear as the next available drive in your system.
- Format the SSD using the DOS format command.

NOTE!	If you wish to make the SSD bootable, you must format it using the /s switch of the format command. Refer to your DOS manual for more information on format.

- If you wish to boot from the SSD make it diskette A.

Installing a Conventional SSD using SRAM or NOVRAM

When the Solid State Disk is composed of SRAM or NOVRAM, it appears as a read/write disk and may be read and written using normal DOS disk commands.

To install an SRAM or NOVRAM SSD, you should follow this procedure:

- Apply power to the cpuModule and run Setup.
- Set **SSD socket 1 or 2** to the appropriate SRAM or NOVRAM device type.
- Set **SSD Window** to a value which will not conflict with other hardware or software.

NOTE!	The memory window selected for DOC must not be used by any other program or hardware device. Make sure this window is not in use by EMM386 or another memory manager, or an Ethernet card, PCMCIA card, etc.

- Save your changes and exit Setup.
- Turn off the cpuModule.
- Install the memory device into the cpuModule socket.
- Reboot the cpuModule. The SSD should appear as the next available drive in your system.
- Format the SSD using the DOS format command.

If you wish to make the SSD bootable, you must format it using the /s switch of the format command. Refer to your DOS manual for more information on the format command.

If you wish to boot from the SSD make it diskette A.

Notes on Formatting an SRAM or NOVRAM SSD

For most purposes, you can format the SRAM or NOVRAM SSD as you would a 1.44MB floppy, using a command line similar to:

for example *format b: /u*

ROM-DOS™:*format b: /u /n:9 /t:40*

See your DOS manual for details on the format command and its parameters.

Installing a Conventional SSD using EPROM or Flash other than Atmel

EPROMs or Flash EPROMs for an SSD must be programmed externally to the cpuModule, using an EPROM programmer and following the procedure below.

To install an EPROM SSD, you should follow this procedure:

- Apply power to the cpuModule and run Setup.
- Set **SSD socket 1 or 2** to the appropriate EPROM or Flash type.
- Set **SSD Window** to a value which will not conflict with other hardware or software.

NOTE!	The memory window selected for DOC must not be used by any other program or hardware device. Make sure this window is not in use by EMM386 or another memory manager, or an Ethernet card, PCMCIA card, etc.

- Save your changes and exit Setup.
- Boot the cpuModule.

After booting, the Solid State Disk will be seen by the system as a write-protected floppy. DOS commands normally used to read floppy disks will work with the SSD.

Directly Accessing the Solid State Disk

If you wish to directly access the Solid State Disk of the cpuModule contact the factory for additional information on doing so.

CHAPTER 6: HARDWARE REFERENCE

This appendix gives information on the cpuModule hardware, including:

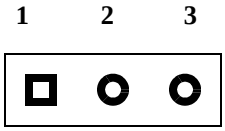
- jumper settings and locations
- solder jumper settings and locations
- mechanical dimensions
- processor thermal management

Jumpers and Solder Jumper Settings

Many cpuModule options are configured by positioning jumpers. Jumpers are labeled on the board as **"JP"** followed by a number.

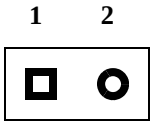
Some jumpers are three pins, allowing three settings:

- pins 1 and 2 connected (indicated as "1-2")
- pins 2 and 3 connected (indicated as "2-3")
- no pins connected.



Some jumpers are two-pin, allowing two settings:

- pins 1 and 2 connected (indicated as "closed")
- pins 1 and 2 un-connected (indicated as "open")



Solder jumpers are set at the factory and are rarely changed. Solder jumpers are located on the module's solder side and component side

The figures below shows the locations of the jumpers and solder jumpers used to configure the cpuModule. To use the figure, position the module with the PC/104 bus connector at the six o'clock position and the component side facing up. The table below lists the jumpers and their settings.

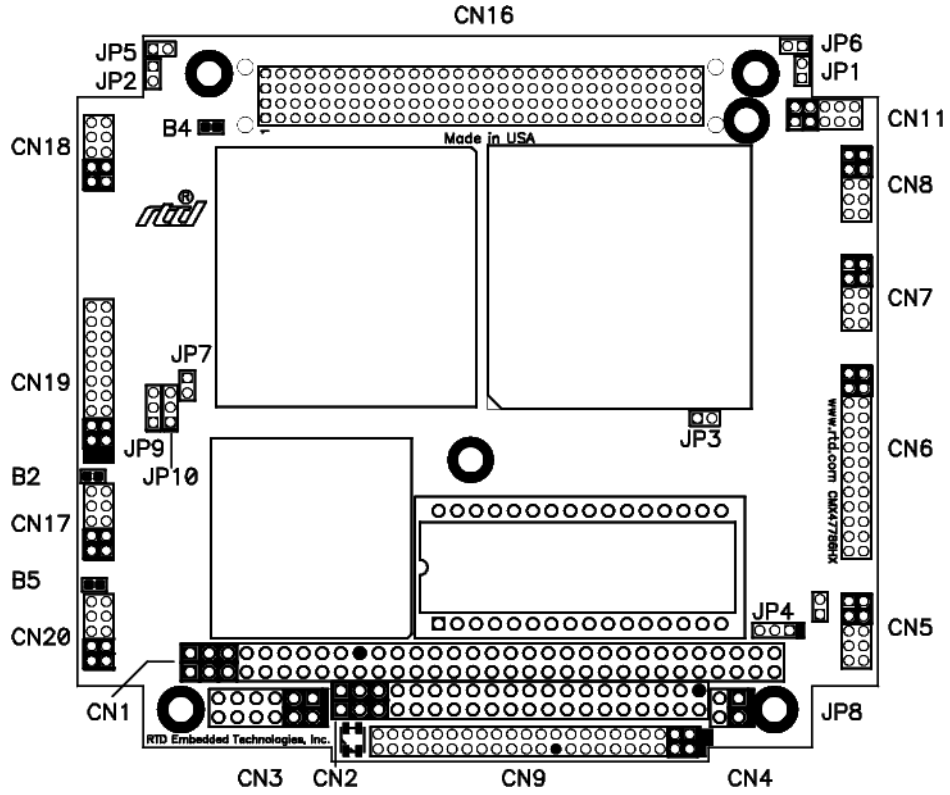


Figure 4: CMX47786HX Jumper Locations - Top

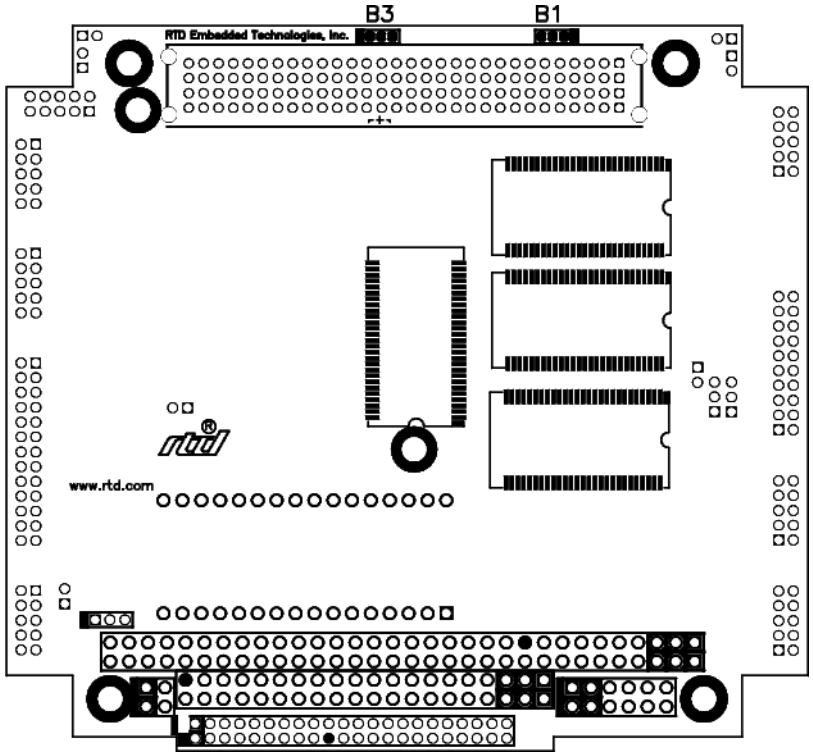


Figure 5: CMX47786HX Jumper Locations - Bottom

Table 36: Jumpers and Default Settings

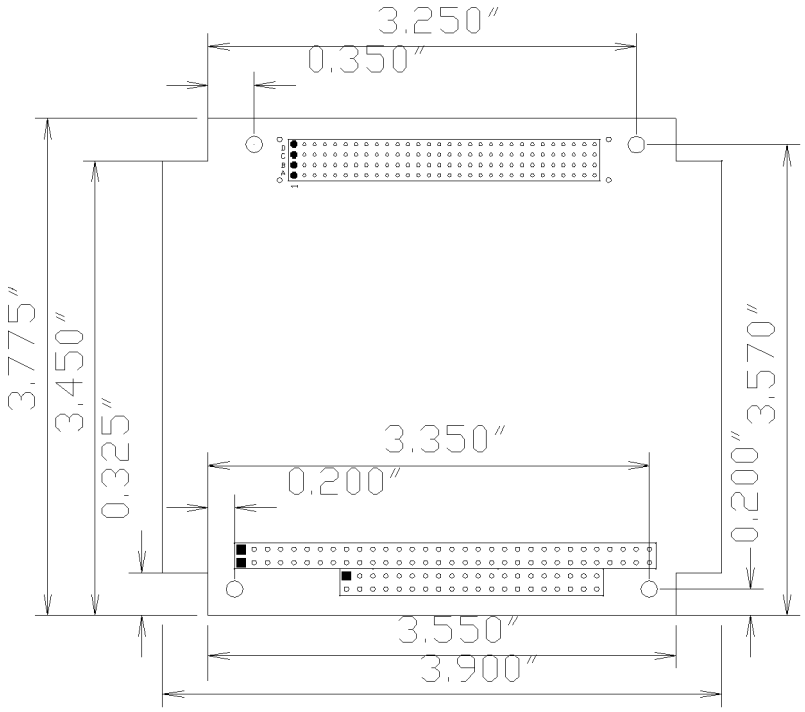
JP1	2-pin jumper Used to enable/disable 120 ohm termination resistor on first serial port for RS-422/485 mode. default: Open (no termination)
JP2	2-pin jumper Used to enable/disable 120 ohm termination resistor on first serial port for RS-422/485 mode. default: Open (no termination)
JP4	3-pin jumper Used to select power for an SSD socket. • 1 to 2 is 5 Volts on board. • 2 to 3 is 5 volts on board when power on and battery backup when power off default: Positions 1 and 2.
JP5	2-pin jumper Used for setting to boot to Fail Safe Boot ROM and restoring factory BIOS settings default: Open
JP6	2-pin jumper Factory use only; do not close. default: Open
JP9	3-pin jumper Used to select power for the Flat Panel Backlight • 1 to 2 is 12V • 2 to 3 is 5V
JP10	3-pin jumper Used to select power for the Flat Panel Electronics • 1 to 2 is 3.3V • 2 to 3 is 5V

Table 37: Solder Jumpers and Default Settings

B1	Three position solder jumper to select 3.3 or 5 volt signaling on the PCI bus. • Pads 1-2 shorted is 5 Volt signaling • Pads 2-3 shorted is 3.3 Volt signaling. Default Setting
B2	Two position solder jumper to connect USB shield ground to frame ground. Default is not connected.
B3	Three position solder jumper to select the source of 3.3V on the PCI bus. • 2-3 -- PC/104 *Plus* PCI bus +3.3 volts supplied by on-board +3.3 volt regulator. Maximum current source is two (2) amps. • 1-2 -- PC/104 *Plus* PCI bus +3.3 volts supplied by power connector P9 pins 10 and 12. Default setting.
B4	Two position solder jumper to connect frame ground to digital ground. Default is not connected.
B5	Two position solder jumper to connect Ethernet shield ground to frame ground. Default is not connected.

Mechanical Dimensions

The following figure shows mechanical dimensions of the module. Dimensions are in inches (mm).



Processor Thermal Management

The industrial grade processor IC of the cpuModule must receive adequate cooling to ensure proper operation and good reliability. The case temperature of the processor must not exceed +100°C..

NOTE!	This cpuModule is *not* warranted against damage caused by overheating due to improper or insufficient heatsinking or airflow.

CHAPTER 7: TROUBLESHOOTING

Many problems you may encounter with operation of your cpuModule are due to common errors. This chapter will help you get your system operating properly.

It contains:

- Common problems and solutions
- Troubleshooting a PC/104 system
- How to obtain technical support
- How to return a product

Common Problems and Solutions

The following table lists some of the common problems you may encounter while using your cpuModule, and suggests possible solutions.

If you are having problems with your cpuModule, please review this table *before* contacting technical support.

Problem	Cause	Solution
Windows/Linux does not display entire screen.	Flat panel is enabled, and set to a lower resolution than the operating system resolution	Disable flat panel, or set it to a higher resolution.
cpuModule “will not boot”	no power or wrong polarity	check for correct power on PC/104 bus connectors
	incorrect setup (video disabled, etc.)	install jumper JP5; reboot and press {Del.} key to run Setup
	defective or mis-connected device on bus	check for misaligned bus connectors; remove other cards from stack
	cable connected backwards	verify all cables are connected correctly
	SSD installed backwards	check for an SSD memory installed in socket backwards
does not recognize Setup changes	jumper JP5 installed	turn off power, remove JP5, reboot
will not boot from particular drive or device	device not bootable	use sys command on drive or re-format the device using the **/s** switch
	device not formatted	format drive using **/s** switch
	power not connected to boot drive	connect power cable to floppy or hard drive
Atmel Flash shows disk space available, but it cannot be written	part smaller than 1.44MB was formatted as 1.44MB; it will show space available even when full	ignore “disk space remaining” messages from DOS REMEMBER! A bootable disk contains 3 hidden files plus format info, totalling about 150kB
will not boot from DiskOnChip®	DiskOnChip® is not the only hard drive in system	disable other hard drive(s) in system
	using wrong DiskOnChip® device (not 32 pin)	change to correct (32 pin) DiskOnChip®
	Boot device not set to Hard disk	run Setup and set boot device to Hard Drive

erratic operation	excessive bus loading	reduce number of PC/104 modules in stack; remove termination components from bus signals; remove any power supply bus terminations
	power supply noise	examine power supply output with oscilloscope; glitches below 4.75Vdc will trigger a reset; add bypass caps
	power supply limiting	examine power supply output with oscilloscope; check for voltage drop below 4.75V when hard drive or floppy drive starts; add bypass caps
	temperature too high	add fan, processor heatsink, or other cooling device(s)
	memory address conflict	check for two hardware devices (e.g. Ethernet, SSD, Arcnet, PCMCIA) trying to use the same memory address check for two software devices (e.g. EMM386, PCMCIA drivers, etc.) trying to use the same memory addresses check for hardware and software devices trying to use the same memory address check for an address range shadowed (see Advanced Setup screen) while in use by another hardware or software device
	I/O address conflict	check for another module trying to use I/O addresses reserved for the cpuModule between 010h and 01Fh check for two modules (e.g. dataModules, PCMCIA cards, Ethernet) trying to use the same I/O addresses
keyboard does not work	keyboard interface damaged by misconnection	check if keyboard LEDs light
	wrong keyboard type	verify keyboard is an 'AT' type or switch to 'AT' mode
floppy drive light always on	cable misconnected	check for floppy drive cable connected backwards
two hard drives will not work, but one does	both drives configured for master	set one drive for master and the other for slave operation (consult drive documentation)
floppy does not work	“data error” due to drive upside down	orient drive properly (upright or on its side)

will not boot when video card is removed	illegal calls to video controller	look for software trying to access non-existent video controller for video, sound, or beep commands
COM port will not work in RS422 or RS485 modes	not configured for RS422/485	correctly configure serial port in Setup program
COM port will not transmit in RS422 or RS485 mode	not enabling transmitters	control RTS* bit of Modem Control Register to enable transmitters; see Serial Port descriptions
date and time not saved when power is off	no backup battery	connect a backup battery to the Multifunction connector
cannot enter BIOS	quick boot enabled with no hard drives	install JP5, reboot, and press {Del.} to enter setup.
Interrupts not working for aDIO, Watchdog Timer, or other ISA device.	IRQ conflict with PCI bus.	Reserve interrupt as Legacy ISA in Setup.

Troubleshooting a PC/104 System

If you have reviewed the preceding table and still cannot isolate the problem with your cpuModule, please try the following troubleshooting steps. Even if the resulting information does not help you find the problem, it will be very helpful if you contact technical support.

Simplify the system. Remove items one at a time and see if one particular item seems to cause the problem.

Swap components. Try replacing items in the system one-at-a-time with similar items.

How to Obtain Technical Support

If after following the above steps, you still cannot resolve a problem with your cpuModule, please assemble the following information:

- cpuModule **model**, **BIOS version**, and serial number
- list of **all boards in system**
- list of settings from cpuModule Setup program
- printout of autoexec.bat and config.sys files (if applicable)
- description of problem
- circumstances under which problem occurs

Then contact factory technical support:

Phone: 814 234-8087
Fax: 814 234-5218
E-mail: techsupport@rtd.com

CHAPTER 8: LIMITED WARRANTY

RTD Embedded Technologies, Inc. warrants the hardware and software products it manufactures and produces to be free from defects in materials and workmanship for one year following the date of shipment from RTD Embedded Technologies, INC. This warranty is limited to the original purchaser of product and is not transferable.

During the one year warranty period, RTD Embedded Technologies will repair or replace, at its option, any defective products or parts at no additional charge, provided that the product is returned, shipping prepaid, to RTD Embedded Technologies. All replaced parts and products become the property of RTD Embedded Technologies. Before returning any product for repair, customers are required to contact the factory for an RMA number.

THIS LIMITED WARRANTY DOES NOT EXTEND TO ANY PRODUCTS WHICH HAVE BEEN DAMAGED AS A RESULT OF ACCIDENT, MISUSE, ABUSE (such as: use of incorrect input voltages, improper or insufficient ventilation, failure to follow the operating instructions that are provided by RTD Embedded Technologies, “acts of God” or other contingencies beyond the control of RTD Embedded Technologies), OR AS A RESULT OF SERVICE OR MODIFICATION BY ANYONE OTHER THAN RTD Embedded Technologies. EXCEPT AS EXPRESSLY SET FORTH ABOVE, NO OTHER WARRANTIES ARE EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND RTD Embedded Technologies EXPRESSLY DISCLAIMS ALL WARRANTIES NOT STATED HEREIN. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES FOR MECHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED TO THE DURATION OF THIS WARRANTY. IN THE EVENT THE PRODUCT IS NOT FREE FROM DEFECTS AS WARRANTED ABOVE, THE PURCHASER'S SOLE REMEDY SHALL BE REPAIR OR REPLACEMENT AS PROVIDED ABOVE. UNDER NO CIRCUMSTANCES WILL RTD Embedded Technologies BE LIABLE TO THE PURCHASER OR ANY USER FOR ANY DAMAGES, INCLUDING ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES, EXPENSES, LOST PROFITS, LOST SAVINGS, OR OTHER DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT.

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR CONSUMER PRODUCTS, AND SOME STATES DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

RTD Embedded Technologies, Inc.
103 Innovation Blvd.
State College PA 16803-0906
USA
Our website: www.rtd.com